

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Conception d'un support informatique d'aide à l'enseignement d'une méthodologie homéopathique

Gillet, Françoise

Award date:
1984

Awarding institution:
Université de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

ANNEE ACADEMIQUE 1983 - 1984

CONCEPTION D'UN SUPPORT INFOR-
MATIQUE D'AIDE A L'ENSEIGNEMENT
D'UNE METHODOLOGIE HOMEOPATHIQUE

Promoteur: J. FICHEFET

Mémoire présenté pour l'obtention
du grade de Licence et Maîtrise
en Sciences Informatiques par
GILLET Françoise

Je tiens ici à remercier particulièrement Monsieur J. FICHEFET, promoteur de ce mémoire, pour ses précieuses indications et ses encouragements.

Je remercie également Monsieur le Docteur JACQUES qui a guidé mes premiers pas dans la discipline homéopathique, de même que Messieurs GARDIN et PARIS dont l'aide et les conseils ont permis de mener à bien cette étude.

T A B L E D E S M A T I E R E S

=====

INTRODUCTION	1
PREMIERE PARTIE : CONTEXTE DU PROJET	2
I. THERAPEUTIQUE HOMEOPATHIQUE DANS LA PRATIQUE MEDICALE	3
I.1. Allopathie et homéopathie	3
I.2. Les principes de l'homéopathie	5
I.3. Les principaux outils de l'homéopathie	9
I.4. La démarche homéopathique	11
1. Première prescription	11
a) Objectif	11
b) Relevé des symptômes	
Diagnostic homéopathique	
Individualisation du malade	11
c) Diagnostic remédial	
Individualisation du remède	14
2. Pronostic - Evolution de la maladie	14
3. Seconde prescription	15
II. LE PROJET INFORMATIQUE COORDONNE PAR l'U.I.H.N. DANS LE CADRE d'ARCHIMEDE	16
II.1. Présentation du projet	16
II.2. Base des données informatisées	17
II.3. Traitements informatisés	17
III. LES TYPES d'OUTILS d'AIDE AU DIAGNOSTIC	21
III.1. Outils empiriques d'aide au diagnostic	21
III.2. Méthodes d'aide au diagnostic	21

DEUXIEME PARTIE : UN SYSTEME D'AIDE A L'ENSEIGNEMENT D'UNE METHODOLOGIE HOMEOPATHIQUE	24
I. L'ENSEIGNEMENT PAR ORDINATEUR	25
I.1. Evolution des concepts en pédagogie expérimentale	25
I.11. Apports de la psychologie	25
I.12. Pédagogie expérimentale et enseignement programmé	26
I.13. Apports des technologies nouvelles	27
I.14. Evolution dans la conception des didacticiels: les objets pédagogiques	32
I.2. Quelques applications de l'enseignement assisté par ordinateur dans le domaine de l'aide au diagnostic médical	36
I.3. Rédaction des didacticiels	39
II. SYSTEME D'AIDE A L'ENSEIGNEMENT DU DIAGNOSTIC HOMEOPATHIQUE	43
II.1. Présentation du projet	43
II.11. Objectifs	43
II.12. Caractéristiques pédagogiques	43
II.13. Poste de travail	44
II.14. Conception du système Spécification des objets pédagogiques	45
II.2. Analyse fonctionnelle	55
II.21. Description et spécification des données	55
II.21.1. Description des données	55
II.21.2. Spécification des données	71
II.22. Spécification des traitements	118
II.3. Choix d'implémentation	175
II.31. Choix du langage de programmation	175
II.32. Choix d'une organisation de fichiers	176
CONCLUSION	194
ANNEXE : STRUCTURES AUXILIAIRES	195

INTRODUCTION

Progressivement, les résistances manifestées envers l'enseignement assisté par ordinateur s'estompent, essentiellement dans des domaines limités et précis où les concepts peuvent être clairement décomposés et structurés.

Les efforts consacrés au développement de méthodologies expérimentales de l'enseignement ont permis de mettre en évidence :

- des modèles de conception de logiciels de plus en plus souples, extensibles, orientés vers des tâches pédagogiques,
- ainsi que des stratégies d'enseignement de plus en plus tournées vers la prise d'initiative de la part de l'étudiant.

Les apports des expériences en EAO peuvent également éclairer la démarche de conception d'un outil d'aide à l'enseignement d'une méthodologie homéopathique.

Dans notre projet, le domaine particulier à enseigner est caractérisé par une diversité des sources d'information, de connaissances formelles ou informelles, des outils de décisions, et est voué à évoluer, à s'étendre à d'autres données et modèles.

Le projet de ce mémoire vise essentiellement à moduler et à paramétrer un outil destiné à concevoir, structurer et exécuter des leçons élaborées par des médecins rompus à la discipline homéopathique.

P R E M I E R E P A R T I E

C o n t e x t e d u p r o j e t

I. THERAPEUTIQUE HOMEOPATHIQUE DANS LA PRATIQUE MEDICALE.

I.1 - Allopathie et homéopathie.

L'attrait exercé par les médecines "parallèles" depuis quelques années dans nos pays occidentaux met en évidence un certain malaise vis-à-vis de la médecine officielle, allopathique.

Celle-ci tend de plus en plus à devenir :

- technicienne, de par le développement prodigieux et la sophistication des outils d'analyse, d'examen, d'observation, d'intervention chirurgicale;
- spécialiste, de par sa méthode diagnostique et thérapeutique: seul est pris en considération l'organe malade, les indices révélateurs exclusifs de la maladie. Elle n'accorde que peu d'intérêt ou néglige les symptômes plus subjectifs présentés par le patient, l'histoire pathologique de celui-ci, son contexte familial, socio - professionnel ...
- impuissante dans les cas de troubles d'origine fonctionnelle où aucun symptôme objectif, résultat de la maladie, n'est encore apparu de façon précise.

Sa méthode diagnostique est basée sur la détermination, grâce aux connaissances nosologiques les plus modernes, d'un commun dénominateur dans l'ensemble des symptômes objectifs présentés par différents patients souffrant des mêmes troubles organiques.

Sa technique thérapeutique est régie par la loi des contraires "contraria contrariis curantur".

Le diagnostic remédial est immédiat dès lors que la maladie a été reconnue, étiquetée.

L'unité vitale, clinique et thérapeutique considérée en allopathie est l'organe malade, elle ne tient compte que des aboutissants de la maladie.

L'homéopathie aborde le problème de la santé et de la maladie de manière plus globale. L'unité vitale, clinique et thérapeutique devient l'homme :

- dans son intégrité: sa personnalité physique, psychique, conditionnée par son tempérament, sa constitution, son terrain.
- dans son environnement: familial, professionnel, ...
- dans son histoire : prise en compte de l'ensemble de la biopathographie.

Sa méthode diagnostique ne se base pas d'abord sur les aboutissants de la maladie, mais avant tout sur l'étude de l'origine des troubles dont se plaint le patient, la détermination de la manière dont celui-ci "vit" sa maladie, car cette dernière est modulée par le contexte global (terrain, environnement, histoire); ainsi sa constitution, son tempérament ou tout agent perturbateur extérieur peut provoquer des troubles fonctionnels avant que la "maladie" n'atteigne les organes eux-mêmes; ces troubles sont donc autant de signes à prendre en considération (tout être vivant est un réactif autrement plus sensible que tout appareil de laboratoire même perfectionné).

Le diagnostic remédial n'est donc pas immédiat dès lors que la maladie a été étiquetée, il faut encore relever les symptômes particuliers au malade et déterminer le remède le plus adéquat à l'ensemble de ces signes afin de prescrire le minimum nécessaire et non le maximum tolérable, pour rééquilibrer l'ensemble de l'organisme.

La technique thérapeutique, en homéopathie, est régie par la loi des semblables: "similia similibus curantur".

Dans la mesure où les lésions organiques ne sont pas trop importantes, où le point de non retour n'est pas atteint, l'homéopathie peut jouer un rôle curatif en profondeur, voire préventif aux yeux de ceux qui ne considèrent comme symptômes morbides que les aboutissants de la maladie.

I.2- Les principes de l'homéopathie

La loi qui régit l'homéopathie en tant que technique thérapeutique, et lui donne son nom, est celle de similitude :

('ομοιο - παθής : qui a les mêmes affections).

Cette loi, décrite par Hippocrate, a été exploitée de façon empirique depuis l'Antiquité, mais elle n'a été expérimentée de façon systématique qu'à la fin du XVIIIe siècle avec Hahnemann, à partir du moment où celui-ci a fait l'expérience sur lui-même qu'une substance pouvait provoquer sur un homme sain les mêmes symptômes morbides que ceux qu'elle était capable de guérir.

Autour de cette loi, principe essentiel, ont pu être dégagés six autres principes; ils ont été recensés et détaillés par Hahnemann dans son ouvrage "L'Organon de l'art de guérir".

L'observation de la nature fait apparaître que celle-ci, considérée dans son ensemble, possède toutes les ressources nécessaires pour lutter, afin de retrouver son équilibre, contre des perturbations provoquées par certains de ses éléments: principe de la force médicatrice de la nature (natura medicatrix), mais lorsque le patient n'a plus la force de lutter contre la maladie, lorsque son "énergie vitale" est devenue insuffisante, il devient nécessaire de trouver un remède dont le tableau symptomatologique, déterminé par expérimentation pure sur des hommes "sains", ressemble le plus au tableau morbide clinique déterminé par l'examen minutieux des origines et du mode de développement de la maladie du patient ainsi que de son terrain constitutionnel (miasmes - diathèses) qui module son mode de réaction aux agents perturbateurs (individualisation du malade et du remède, loi de similitude). Ce remède représente la maladie de substitution correspondant à celle du patient, capable de lui venir en aide contre l'agent déclencheur de son déséquilibre physique ou psychique. Son action sera d'autant plus efficace et moins nocive qu'il aura été dilué et dynamisé (homogénéisé) : loi infinitésimale.

Ces différents principes sont repris séparément ci-après :

1) Loi de similitude

La loi de similitude est la loi naturelle qui régit l'homéopathie.

" Une maladie dynamique est vaincue et détruite de façon durable par une autre maladie plus forte et celle-ci, sans être de même nature, lui ressemble beaucoup dans ses manifestations" ([13] § 26).

La recherche de ce "similimum", maladie de substitution, doit se baser sur l'ensemble des symptômes physiques, psychiques, présentés par le patient, afin que ce remède puisse renforcer toutes les réactions de l'organisme en lutte (ne tenter de faire disparaître que les symptômes organiques pourrait détruire un système de défense de l'organisme qui est en train de réagir à un agent perturbateur).

2) Force médicatrice de la nature

" La maladie primitive est curable d'emblée et spontanément par l'action de la seule force de l'organisme" ([13] p. 3).

Si un individu n'a plus la force nécessaire pour rétablir un état d'existence harmonieux, un équilibre, il faut tenter de renforcer les réactions naturelles. La recherche du similimum s'inscrit donc dans une recherche plus fondamentale d'imiter la manière dont la nature arrive à s'équilibrer.

3) Dynamisme vital

La vie de tout être humain se définit en termes d'énergie, force qui lui permet d'assurer sa permanence, tant sur le plan organique que sur le plan fonctionnel. L'état de santé est un état d'existence complet déterminé par une harmonie de cette énergie vitale : l'organisme sain est capable de réagir et de rétablir son équilibre lorsqu'un agent perturbateur vient provoquer des troubles. L'état de maladie est également un état d'existence complet qui se traduit par un affaiblissement du dynamisme vital; ses réactions, symptômes morbides, sont signes de l'effort non suffisant fourni pour retrouver son état d'équilibre. Cette énergie vitale se dégrade peu à peu, jusqu'à atteindre le point de non retour possible à l'état harmonieux.

4) Doses infinitésimales

Ce principe est indissociable de la loi de similitude: le similimum ayant été sélectionné, il faut pouvoir en déterminer la dose adéquate au cas du patient.

Comme l'absorption d'une substance de base entraîne souvent une aggravation des symptômes (à moins qu'elle n'ait aucun effet), on procède à des dilutions successives de cette drogue dans un solvant servant de véhicule aux propriétés pharmaco-dynamiques de la substance (eau distillée, alcool, ...) et à sa dynamisation (en secouant le mélange à chaque étape de la dilution) pour en assurer son homogénéité.

Cependant, étant donné la loi d'Avogadro, il n'existe plus de molécules de la substance dans les préparations de haute dilution, mais on émet l'hypothèse que l'agencement des ions moléculaires du solvant est modifié grâce à ce procédé, on pourrait alors observer un spectre propre à chaque remède dilué. Plus ce procédé de dilution-dynamisation est appliqué, plus les propriétés physico-chimiques du remède se manifestent.

5) Expérimentation pure

Des expériences empiriques ont mis en évidence que des substances sont capables de guérir des symptômes analogues à ceux qu'elles peuvent produire. Afin de systématiser cette découverte et de déterminer les propriétés de différentes drogues et les symptômes qu'elles provoquent (symptômes dits pathogénétiques), on les expérimente :

- sur des personnes saines, c'est-à-dire ressentant un bien-être physique, mental, social, afin d'éviter au maximum des superpositions de tableaux symptomatologiques,
- chacun séparément, afin de créer une image pathogénétique de chaque drogue.

(Mais on pourrait expérimenter également des complexes de remèdes - déjà expérimentés séparément - sur des personnes saines, afin d'élargir l'ensemble des pathogénésies et ainsi pouvoir répondre à un plus grand nombre d'images pathologiques cliniques).

Les résultats des expérimentations pures sont recueillis dans les Matières Médicales; les remèdes y sont classés alphabétiquement. Elles constituent des protocoles toxicologiques sur lesquels se base la recherche du similimum.

Il faut cependant noter que, dans ces protocoles, ne figurent pas les symptômes lésionnels, les remèdes n'étant pas expérimentés à des doses très toxiques.

6) Miasmes - diathèses

Une diathèse est définie comme un état morbide constitutionnel s'exprimant par des altérations structurelles, fonctionnelles et psychiques, transmises héréditairement et manifestées dans l'histoire pathologique d'un individu.

Le terrain constitutionnel conditionne à la fois la sensibilité de l'individu à certains agents morbides particuliers (et le prédispose à contracter certaines maladies), et son mode de réaction vis-à-vis de ces agents.

Les modes de réaction peuvent se définir par rapport à trois axes différents : les trois diathèses de base (ou miasmes, ou modes de réactions de référence) que Hahnemann, à partir des connaissances physico-pathologiques de son temps, a nommés :

- la psore : hyporéaction provenant de certains refoulements
- la sycose : hyperréaction
- la syphilis : réaction destructrice se répercutant sur le plan physique et mental.

Il y a pratiquement autant de combinaisons nuancées de ces trois diathèses de base qu'il y a d'individus. Ainsi plusieurs personnes manifestant des symptômes morbides communs réagiront différemment selon la modulation induite par leur terrain miasmatique.

7) Individualisation du malade et du remède

L'individualisation du malade consiste en la mise en évidence de son mode particulier de réaction aux agressions, de son état diathésique, de son terrain miasmatique personnel, par le relevé de tous les symptômes rares, d'une intensité singulière, tant sur le plan psychique que fonctionnel, lésionnel. On obtient ainsi une image complète de l'état d'existence du patient à un instant donné. Le remède unique (dans la mesure où des complexes de remède n'ont

pas encore fait l'objet d'expérimentation pure) correspondant au malade, doit posséder l'image pathologique la plus ressemblante à celle du patient.

I.3- Les principaux outils de l'homéopathie

1) L'Organon de l'art de guérir

Cet ouvrage d'Hahnemann recueille les principes mêmes de l'homéopathie, énonce les règles de base de l'art de guérir.

2) Les Matières Médicales

Les Matières Médicales recueillent les ensembles des pathogénésies de différentes substances, ou protocoles toxicologiques, classées alphabétiquement.

Chacune de ces pathogénésies recense un maximum de symptômes induits par la drogue en question sur une personne saine lors d'une expérimentation pure. Ces symptômes sont classés alphabétiquement en 31 chapitres correspondant aux différentes subdivisions du corps humain.

La première Matière Médicale est celle de Hahnemann et regroupe plus de 1500 remèdes. D'autres sont venues la compléter, l'enrichir. Parmi les plus connues, citons celles de Allen, de Boericke.

3) Les répertoires

La manipulation des Matières Médicales, dont l'accès s'effectue sur base du nom du remède, étant fastidieuse, des répertoires de symptômes ont été créés. Chaque symptôme référence le ou les remèdes qui l'ont guéri ou qui le contiennent dans leur pathogénésie.

Hahnemann avait déjà créé un répertoire à partir de sa Matière Médicale. Parmi les plus connus actuellement, citons celui de Kent, celui de Barthel.

Le répertoire de Kent est écrit en anglais et structuré de façon arborescente :

- il est subdivisé en 31 chapitres selon les différentes parties du corps humain
- chacune de ces subdivisions se compose de différentes rubriques correspondant à un symptôme général, classées alphabétiquement selon la langue anglaise.

- chaque rubrique contient essentiellement :
 - . une liste générale des remèdes qui ont produit ce symptôme
 - . une liste de remèdes selon les moments de la journée
 - . une liste de remèdes selon les modalités (circonstances d'aggravation) de ce symptôme.

(Si le symptôme est une douleur, d'autres listes de remèdes viennent s'ajouter, correspondant à l'irradiation, la localisation, le caractère de la douleur, ...).

Tous les remèdes apparaissant dans ces différentes listes sont affectés d'un degré. Le degré d'un remède dans une liste particulière associée à un symptôme, détermine une estimation qualitative de l'efficacité de ce remède dans la guérison de ce symptôme.

Il existe trois degrés distincts pour un remède donné dans une liste par rapport au symptôme correspondant :

- a) presque tous les individus sains ayant participé à l'élaboration de la pathogénésie de cette substance ont présenté ce symptôme. Son efficacité thérapeutique a été confirmée cliniquement.
- b) 40 à 50 % des individus sains ont présenté ce symptôme lors d'une expérimentation pure. Son efficacité a été vérifiée également sur des cas cliniques.
- c) quelques individus ont présenté ce symptôme lors d'une expérimentation pure. De plus, certains malades ont été guéris de ce symptôme alors que celui-ci ne figurait pas dans le protocole d'expérimentation pure de ce remède.

Le répertoire de Barthel et Klunker est structuré de la même façon que celui de Kent. Il donne, en plus de l'expression des symptômes en anglais, leur traduction en français et en allemand.

4) Les expérimentations cliniques

L'expérimentation pure ne pouvant s'effectuer jusqu'à provoquer des symptômes lésionnels, les Matières Médicales ne sont donc pas toujours suffisantes. Le relevé statistique des symptômes guéris par les différents remèdes est donc une source de renseignements précieuse.

Les "symptômes guides" de Hering mettent en évidence les symptômes les plus régulièrement guéris par un remède prescrit.

5) Toxicologie

La toxicologie est une source supplémentaire symptomatologique. Aiguë, chronique, volontaire, professionnelle, iatrogène ou accidentelle, elle renseigne des symptômes très violents, aboutissant rapidement à des lésions organiques (celles-ci pouvant éclipser les nuances symptomatologiques).

I.4- La démarche homéopathique

1. Première prescription

a) Objectif

Il s'agit de déterminer le remède le plus probable à un moment donné où l'on "flasche" l'état énergétique déficient du patient.

b) Relevé des symptômes, diagnostic homéopathique: individualisation du malade.

- Anamnèse

Dans un premier temps, le médecin recueille les renseignements personnels, les antécédents médicaux (opérations, maladies, médicaments déjà pris, ...), les renseignements familiaux, afin de déceler la nature du terrain constitutionnel du patient.

- Examens cliniques

En ce qui concerne l'affection pour laquelle le patient vient consulter le médecin, certains examens somatiques sont systématiques. Les analyses et examens déjà subis par le patient sont inclus dans le dossier.

- Examen homéopathique

. Interrogation

Si l'interrogation reste un art, il existe cependant certains principes à respecter lors de l'interrogation :

- " écouter, transcrire, questionner, coordonner" sont les quatre principes relevés par P. Schmidt dans son ouvrage "L'art d'interroger"
- ne pas poser des questions directes ou contenant une alternative, ou influençant la réponse du patient.
- Ubi - Quando - Quomodo - Cur - Quid - Aliis, sont des mots-clés autour desquels s'articule l'interrogatoire afin de déterminer l'étiologie de la maladie.

L'expérience peut faire apparaître également l'efficacité de certains tests propédeutiques dans des cas précis.

. Répertorisation

Transcription des symptômes relevés en termes répertoriés

. Sélection

Parmi tous les symptômes recueillis, il faut pouvoir sélectionner ceux qui sont les plus discriminants dans le choix du remède, les plus particuliers ou étranges (syndrome minimum de poids maximum). Aucune règle systématique ne peut s'appliquer pour sélectionner ce syndrome si ce n'est l'expérience et une connaissance approfondie des Matières Médicales.

. Hiérarchisation des symptômes

Tous les symptômes n'ont pas la même importance dans l'élection du remède. Un symptôme qui se manifeste de façon incontrôlée ou inconsciente a plus de valeur que les symptômes dont le patient se plaint, car ces derniers ont souvent déjà fait l'objet d'une interprétation de la part du malade lui-même et ne sont qu'une manifestation aiguë de la réaction globale, qu'une conséquence d'altérations fonctionnelles plus générales. Kent a hiérarchisé les symptômes pouvant être présentés par un patient selon les niveaux suivants :

- Etiologie manifeste

Ce sont les facteurs déclencheurs ou causes occasionnelles de la maladie qui ne sont pas des symptômes pathogénétiques ni homéopathiques au malade, ils sont à l'origine de la superposition d'une pathologie aiguë ou devenue chronique au terrain du patient.

- Symptômes mentaux

Symptômes rationnels et émotionnels avec une priorité à ces derniers car ils expriment mieux les réactions naturelles.

- Symptômes rares, singuliers, originaux

Ils expriment comment le malade vit de façon particulière sa maladie

Ces trois premiers types de symptômes ne se trouvent à ce niveau de l'échelle de hiérarchisation que s'ils sont clairs, indiscutables

- Symptômes généraux
dont les causes sont incontrôlables. Ils concernent les réactions, les tendances générales de l'individu (amélioration, aggravation, ...) et les modalités de ces réactions (périodicité, latéralité, irradiations,) face aux conditions et modifications des conditions de son environnement (naturel ou social, affectif), à différents moments ou périodes du jour, de l'année, de la vie...
- Intolérances, aversions et désirs
alimentaires ou physiques, ainsi que leurs répercussions; les aversions sont considérées comme plus significatives (moins contrôlées) que les désirs.
- Rêves, sommeil
La qualité du sommeil, la position, les gestes, sont autant de signes des expressions mentales incontrôlées.
- Comportement sexuel et génital

Les niveaux suivants sont souvent utilisés pour affiner ou confirmer un diagnostic :

- Ecoulements (excrétions et sécrétions)
- Symptômes régionaux
- Symptômes pathognomoniques
Symptômes communs à tous les patients souffrant d'une même maladie.
- Symptômes antérieurs
qui réapparaissent après un certain temps.

c) Diagnostic remédial : individualisation du remède

Après le relevé des symptômes qu'il juge significatifs, le médecin homéopathe dispose de plusieurs outils pour déterminer le "similimum" :

- Etude du répertoire

- . Le ou les remèdes qui apparaissent le plus souvent, avec un degré d'efficacité le plus élevé, dans les différentes listes de remèdes associées aux symptômes sélectionnés, servent de base à une analyse plus approfondie.
- . Les remèdes qui se retrouvent seuls dans la liste d'un symptôme de haut niveau de hiérarchisation sont généralement de bons guides.

- Etude des Matières Médicales

Les remèdes retenus dans une première étape font l'objet d'un diagnostic différentiel sur base des Matières Médicales.

- Etude des cas cliniques

Un remède peut être confirmé par l'étude des symptômes "guides" ou combinaisons de symptômes généralement guéris par ce remède

Ces différentes recherches peuvent amener le médecin à reviser le diagnostic homéopathique afin de déterminer le similimum.

2. Pronostic - évolution de la maladie

a) Observations

Dès son absorption, un remède commence à agir, à opérer des changements dans l'organisme, à déclencher des modifications dans les symptômes :

- modifications de fréquence, d'intensité ou de nuance: on peut assister à une amélioration ou à une aggravation objective et/ou subjective du patient.
- certains symptômes disparaissent, d'autres apparaissent.
- l'ordre de ces apparitions et disparitions a de l'importance dans l'analyse de l'évolution de la maladie (cfr. infra loi de Hering).
- modification de la direction générale prise par les symptômes.

Cette analyse est alors affinée par l'étude des modalités, des localisations, horaires, durées de ces changements.

Les différentes orientations que peut prendre l'évolution de la maladie selon l'intensité, la durée, voire la périodicité des temps d'aggravation et d'amélioration, sont décrites dans ([17]).

b) Loi de Hering (ou loi de direction)

Une maladie qui est en voie de guérison :

- évolue de façon centrifuge à partir des centres vitaux vers la périphérie.
- évolue de haut en bas.
- voit disparaître ses symptômes dans l'ordre inverse de leur apparition.

3. Deuxième prescription

Une deuxième prescription devient nécessaire lorsque le premier remède n'agit plus ou que son action est manifestement inadéquate. Après avoir reconsidéré le dossier du patient et étudié l'évolution de la maladie depuis la dernière prescription efficace, le médecin pourra prescrire :

- le même remède, avec éventuellement une autre dynamisation si d'anciens symptômes réapparaissent, ou si l'état du patient reste stationnaire.
- un nouveau remède s'il y a apparition de nouveaux symptômes :
 - . dans le cas où les symptômes prennent une mauvaise direction, le médecin peut prescrire un antidote
 - . il peut prescrire un remède complémentaire pour agir sur l'état constitutionnel du malade après une phase aiguë (remède de crise et remède de fond sont complémentaires).
 - . il peut aussi prescrire un tout autre remède s'il veut changer de tactique de traitement.

+++++

II. LE PROJET INFORMATIQUE COORDONNE PAR L'U.I.H.N. DANS LE CADRE D'ARCHIMEDE

II.1- Présentation du projet.

L'Universitas Internationalis Hominum Novorum (U.I.H.N.) regroupe des médecins homéopathes, acupuncteurs et phytothérapeutes désireux d'approfondir la recherche et de favoriser le développement et l'enseignement de haut niveau dans leur branche respective, avec obtention d'un diplôme au bout de quatre ans d'études.

Dans le cadre de l'Association pour la ReCHerche en Informatique et MEDECine (ARCHIMEDE), un projet informatique a été élaboré concernant l'homéopathie et coordonné par l'U.I.H.N. Il est également prévu d'intégrer au système des données et des modèles associés à la pratique de l'acupuncture.

Le projet en cours a essentiellement pour objectif

- d'apporter une aide humanitaire aux médecins pratiquant la thérapeutique homéopathique, tant dans les pays en voie de développement que dans les pays industrialisés
- de contribuer à poser les bases d'une recherche fondamentale
- de contribuer à poser les bases pour l'élaboration d'un enseignement moderne de l'homéopathie.

Ce projet peut être considéré sous un angle cybernétique ([18]) D'une part, les Matières Médicales et les Répertoires constituent l'organisation de départ des connaissances théoriques: celles-ci sont confrontées à la pratique clinique des médecins; d'autre part, l'expérience même des médecins contribue à approfondir les connaissances homéopathiques. Cette boucle, qui a pour objectif la guérison du malade, a un esprit créatif qui contribue à l'enrichissement réciproque entre l'observateur et l'observation.

Le système global peut se décomposer en deux sous-systèmes qui interagissent.

Sous-système I

Le patient (l'émetteur) et le médecin (le récepteur) dialoguent par le canal de la parole. Le médecin fait appel aux Matière Médicales, Répertoires, symptômes guides, schémas mentaux - la mémoire du système - afin d'individualiser le patient et le remède selon la loi

de similitude. La mémoire du système s'enrichit d'une nouvelle expérience.

Sous-système II

Le médecin (l'émetteur) prescrit le similimum par le canal d'une ordonnance que le malade présente au pharmacien homéopathe (le récepteur). Celui-ci fait appel aux règles de l'homéopathie décrites par Hahnemann pour préparer le remède prescrit - mémoire du système.

II.2- Bases des données informatisées.

Sous-système I.

- Base des données homéopathiques.

Cette base a été obtenue par fusion des répertoires (Kent, Barthel et Klunker) et regroupe environ 60.000 symptômes et 1500 remèdes.

- Matières Médicales.

- Base des dossiers cliniques centralisée.

- Base des données statistiques.

Sous-système II.

- Base des cartes d'identité des substances de base.

II.3- Traitements informatisés.

Sous-système I.

a) En_cours_de_réalisation

- Gestion des données homéopathiques basées sur l'expérimentation pure

Elle offre un outil informatique d'interrogation très souple de la base de données homéopathiques auquel le médecin peut faire appel lors de l'élaboration du diagnostic homéopathique et remédial d'un patient. L'interrogation des Matières Médicales est également prévue.

- Gestion des données cliniques. ([7])

Les dossiers médicaux formalisés, standardisés, fournis par les médecins de l'U.I.H.N. sont insérés dans une base de données cliniques propre à chacun de ces médecins. Une commission de contrôle aura la charge d'étudier les dossiers clôturés et de décider de sa centralisation dans une base de données centrale, en vue de traitements statistiques ou pouvant servir de base à l'enseignement.

- Modèles de connaissances : aide au diagnostic médical.

Les répertoires sont en eux-mêmes un modèle généralisé: à chaque symptôme correspond une liste de remèdes qui contiennent ce symptôme dans leur pathogénésie, ou qui ont guéri ce symptôme lors d'expérimentations cliniques. Cette relation symptôme-remèdes forme une règle de décision. Lors d'une consultation, le médecin relève plusieurs symptômes et obtient donc, après répertorisation, plusieurs règles de décision. Il existe plusieurs modèles qui permettent de les agréger et de proposer un remède ou un classement de remèdes.

Les modèles des fréquences et des degrés déjà utilisés manuellement par les médecins, de même que d'autres modèles d'aide à la décision multi-critères (ELECTRE II par exemple) ont été informatisés (cfr. infra ADM méthodes de sélection de remèdes).

- Aide à l'enseignement du diagnostic médical

Un système d'aide à l'enseignement informatisé est proposé afin d'initier les étudiants à la technique répertoriale de Kent, à la pratique du diagnostic et de la thérapeutique homéopathique, à la manipulation des modèles.

Ce système interactif conversationnel doit permettre de faire appel aux différents modèles et bases de données existants. Il peut être également le support d'un enseignement particulier simulant des cas réels, mettant en évidence les écueils classiques dans les différentes phases diagnostiques.

b) Recherche et développement en ce qui concerne :

- l'expérimentation pure

- . La base de données homéopathiques est utilisée afin d'étudier sa cohérence logique et ses lacunes.
- . L'intégration de nouvelles expérimentations pures permettra d'enrichir la base de données homéopathiques.

- l'expérimentation clinique

- . Les dossiers cliniques pourraient être complétés par de nouveaux renseignements.
- . Les règles de centralisation des dossiers devront être déterminées afin d'assurer la validité des connaissances introduites dans la base de données cliniques.

- une librairie de modèles

- . Etablissement d'états statistiques et validation de la pertinence de ces statistiques par rapport à l'échantillon de référence, et par rapport à leur valeur diagnostique, étant donné que la valeur diagnostique d'un remède n'est pas toujours mise en évidence par une fréquence statistique élevée.
- . Etablissement de symptômes "guides" associés aux remèdes les plus courants.
- . Etablissement de schémas diagnostiques. Ces schémas reflètent les formes les plus typiques des maladies les plus répandues, en termes de symptômes observés et remèdes ayant un certain degré de fiabilité thérapeutique. Ils pourraient être intégrés dans un système expert et prendre la forme :
si tels symptômes sont présents et tels symptômes absents,
alors tel remède doit être prescrit avec un certain degré de fiabilité.

Sous-système II: contrôle de la qualité des remèdes homéopathiques.

- Contrôle des substances de base : gestion des cartes d'identité et méthodologies de réalisation.

A chaque substance de base d'origine minérale, végétale ou animale, doit correspondre une carte d'identité dont la des-

cription systématique comportera des caractéristiques de type biologique, de type physico-chimique. Enregistré sur support informatique, cet atlas pourra être consulté aisément afin de pouvoir comparer la trace de toute substance avec celle de l'étalon muni de sa carte d'identité.

- Contrôle des substances diluées-dynamisées.

Les propriétés à décrire et les techniques physico-chimiques à mettre en oeuvre pour les mettre en évidence reposent encore sur des hypothèses: ce type de contrôle fait donc partie d'un projet à plus long terme.

III. LES TYPES D'OUTILS D'AIDE AU DIAGNOSTIC MEDICAL DANS LE PROJET U.I.H.N.

III.1 - Outils empiriques d'aide au diagnostic.

Ces outils permettent de communiquer au médecin certaines informations tirées soit des répertoires, soit des matières médicales, soit de la base des dossiers cliniques centralisés, sans les modéliser de façon plus ou moins rigide

1. outils concernant la base des données homéopathiques

1) accès sur mots-clé

cet outil favorise le choix d'une interprétation plus précise des symptômes

2) pathogénésies

cet outil favorise une approche plus systématique des matières médicales

3) diagnostics différentiels entre 2 ou 3 remèdes

cet outil permet d'affiner un diagnostic sur base des symptômes propres à chacun des remèdes concurrents

4) présentation du tableau des symptômes et des remèdes

cet outil permet une visualisation claire des règles de dépendance entre remèdes et symptômes.

2. outils concernant les matières médicales

l'accès à différentes matières médicales permet d'enrichir les images pathogénétiques des différents remèdes.

3. outils concernant les dossiers centralisés

quelques primitives d'estimations simples accompagnées de renseignements tels que la taille de l'échantillon pourraient permettre de valider ou d'invalidier l'efficacité de certains remèdes.

III.2 - Méthodes d'aide au diagnostic

Ces méthodes reflètent chacune une démarche particulière et la prise en compte de renseignements bien déterminés lors de

l'établissement du diagnostic remédial

1. modèles associés actuellement à la base des données homéopathiques

1) méthode_des_fréquences

simple et rudimentaire, cette méthode ne prend en considération que le nombre de symptômes, parmi ceux relevés chez un patient, qui sont couverts par un remède, un remède étant préféré à un autre s'il couvre plus de symptômes.

2) méthode_des_degrés

un remède R1 est préféré à un autre R2 si la somme des degrés du remède R1 pour chacun des symptômes qu'il couvre est supérieure à la somme des degrés associée au remède R2.

3) méthode ELECTRE II ([2])

cette méthode prend en considération à la fois les fréquences et les degrés des différents remèdes pour l'ensemble des symptômes couverts; elle permet en plus de pondérer l'importance des symptômes dans la prise de décision

2. méthodes basées sur les données de la base des données cliniques

1) statistiques

objectif :

. on voudrait connaître

- quels sont les symptômes réellement guéris par un remède, leur fréquence absolue dans la base de données, la fréquence conjointe de chaque symptôme et du remède.

- quels sont les complexes de symptômes et leur fréquence absolue et conjointe guéris par un remède, étant donné éventuellement certains facteurs jugés discriminants (la maladie à un stade aigu par exemple...)

- quels sont les remèdes et leur fréquence qui ont guéri certains symptômes (étant donné éventuellement une maladie particulière.

. si la structure et la nature des données de la base des données cliniques le permet, on pourrait vouloir connaître des renseignements concernant l'évolution des symptômes par exemple: dans le cadre d'une maladie ou d'un remède

donné

- tels symptômes apparaissent indiquant que la maladie est en voie de guérison
- tel symptôme disparaît après un temps t

2) schémas diagnostics ([12])

- . sur base de ces statistiques, dans la mesure où l'on dispose d'estimations fiables permettant d'établir des règles de décision pour les formes les plus typiques des maladies les plus courantes, le remède proposé, associé à un complexe de symptômes typiques est celui dont la probabilité si S_i est un complexe de symptômes

R_k est un complexe de remèdes

$P(S_i)$ représente l'estimation de la probabilité de rencontrer le complexe de symptômes S

$P(R_k)$ représente l'estimation de la probabilité de rencontrer le complexe de remèdes R_k

$P(S_i R_k)$ représente l'estimation de la probabilité de rencontrer le complexe symptômes-remèdes S_i et R_k

alors le remède R_k est proposé avec la probabilité

$$P(R_k/S_i) = \frac{P(R_k S_i)}{P(S_i)} = \max_j [P(R_j/S_i)]$$

- . pour les cas atypiques, une adaptation de la méthode ELECTRE II aux données statistiques est envisagée

La liste de ces outils peut s'étendre, les outils peuvent s'affiner; en reflétant un aspect particulier de la prise de décision, il est intéressant de les intégrer dans un système d'aide à l'enseignement afin d'en apprécier la portée et l'utilisation.

DEUXIEME PARTIE

Un système d'aide à l'
enseignement d'une
méthodologie homéopathique

I. L'ENSEIGNEMENT PAR ORDINATEUR

I.1- Evolution des concepts en pédagogie expérimentale

I.11. Apports de la psychologie :

- lois de l'apprentissage et de l'intelligence
- méthodes expérimentales

Les problèmes liés à l'art d'enseigner ne se posent pas d'hier, ils ont suscité beaucoup de réflexions et de théories. Mais c'est seulement depuis les années 1950 que l'on tente, dans ce domaine, de passer du stade artistique des opinions intuitives au stade des expériences systématiques.

Le développement de méthodes systématiques d'enseignement n'a pu se réaliser qu'à partir du moment où la psychologie de l'apprentissage lui-même est passée à un stade expérimental. Parmi les principes de la théorie de l'apprentissage édifiée par Skinner et son école, et appliqués directement dans l'enseignement expérimental, citons :

- un individu apprend et modifie son comportement en observant les conséquences de ses actes.
- le renforcement d'une action est un facteur de stimulation de l'apprentissage; il augmente la durée d'acquisition des nouveaux comportements.
- le renforcement doit suivre immédiatement l'action et doit être fréquent.
- le comportement de l'élève peut être structuré de façon linéaire à partir de petites étapes simples.

Ces principes sont le résultat de méthodes expérimentales basées sur :

- l'étude des comportements d'un sujet en termes rigoureux
- les hypothèses sur les motifs de ces comportements
- la détermination d'un plan d'expérience des variables quantitatives à étudier et des variables à neutraliser momentanément. Les comportements étudiés doivent être mesurables (durée d'apprentissage,

performance à des épreuves de contrôle déterminées...) L'une des variables est utilisée comme critère d'évaluation de la réussite ou de l'échec.

- des méthodes statistiques permettant d'évaluer le degré de certitude des résultats obtenus.

I.12. Pédagogie expérimentale et enseignement programmé

- principes
- stratégies

L'enseignement programmé a introduit l'esprit expérimental dans l'enseignement en se basant sur les lois de l'apprentissage, et surtout sur les méthodes rigoureuses de la psychologie expérimentale. Il a pris son essor aux Etats-Unis à partir de 1958. Les principes essentiels de la pédagogie expérimentale peuvent s'énoncer de la façon suivante :

- Principe de structuration efficace de la matière enseignée :
il est nécessaire d'analyser la matière en composants élémentaires (faits, concepts) et en relations plus ou moins complexes entre ces composants (hiérarchie simple, réseau multiple), en progressant selon la difficulté de la matière.
- Principe d'adaptation :
l'évolution de la présentation de la matière enseignée doit tenir compte des erreurs, difficultés et progrès de l'apprenant.
- Principe de stimulation :
l'apprenant doit être tenu actif par des questions, des sollicitations fréquentes.
- Principe de contrôle en permanence, point par point :
il faut éviter que l'apprenant s'enfonce dans son erreur ou abandonne parce qu'aucun concept ou structure de concepts ne se dégage de ses propres réponses aux différentes sollicitations.

Plusieurs stratégies de programmation des leçons, de structuration des interactions qui caractérisent la situation d'enseignement sont basées sur ces principes :

- Programmation linéaire (modèle Skinnérien) ou modèle des questions-réponses (drill and practice).
 - . la matière est structurée en petites étapes associées à des concepts élémentaires et rapidement assimilables.
 - . les réponses sollicitées sont de type ouvert.
 - . le rôle du renforcement positif est primordial chez Skinner; un programme bien construit ne devrait plus induire d'erreur chez l'élève: la matière est structurée de façon à susciter la bonne réponse, à insérer des rappels, à illustrer les concepts sous toutes ses formes, de telle sorte que l'élève acquière sans difficulté et insensiblement les connaissances enseignées.
- Programmation ramifiée (modèle Crowdérien) ou modèle tutorial.
 - . la matière est structurée en unités d'information plus complexes afin d'éprouver le raisonnement de l'élève.
 - . apprendre, pour Crowder, signifie souvent apprendre à distinguer; les réponses, dès lors, sont de type choix-multiple. Le choix des réponses doit être pensé en fonction des erreurs les plus fréquentes ou les plus graves pouvant être proposées par l'apprenant.
 - . le but de Crowder n'est pas de savoir "comment" l'élève apprend, ni de forger son raisonnement pas à pas en suscitant et renforçant toujours le bon comportement, Crowder cherche plutôt à savoir si l'élève apprend ou non : les unités d'information sont donc construites de façon à établir un diagnostic sur l'état de ses connaissances. L'élève est orienté vers une unité associée au traitement particulier de son erreur.

I.13. Apports des technologies nouvelles

Après une brève apparition des machines à enseigner, dont l'expérience a montré qu'elles n'apportaient aucun avantage sur les manuels d'enseignement programmé et qu'elles ne correspondaient en fait chacune qu'à un seul type de programmation, l'ordinateur a ouvert un avenir prometteur dans le domaine de la pédagogie expérimentale, en transformant quantitativement et qualitativement les

caractéristiques et possibilités de l'enseignement.

a) Diversification des possibilités techniques :

- + le poste de travail est transformé :
 - . dispositifs d'entrée des informations :
 - clavier alphanumérique
 - écran tactile
 - crayons lumineux
 - on tente de développer des dispositifs capables de reconnaître la parole, ...
 - . dispositifs de sortie :
 - écran avec tube cathodique
 - écran de télévision couleur avec des possibilités graphiques
 - écran à plasma
 - imprimante
 - projecteur de diapositives
 - magnétophone
 - vidéophone
 - synthétiseur de la parole, ...
- + les possibilités du poste de travail peuvent être étendues si celui-ci s'intègre dans un réseau de transmission des données (réseau maillé, en étoile, ...)
- + capacités de mémorisation et puissance de calcul :

ces possibilités autorisent des manipulations de volumes de données considérables, la manipulation de structures de données complexes, l'exécution de traitements pouvant être fastidieux et faisant appel aux capacités logiques et de calcul importantes de l'ordinateur.
- + capacité d'utilisation en mode interactif, conversationnel
- + temps de réaction court.

b) Diversification des possibilités logicielles :

- + base de logiciels spécialisés et librairies standards accessibles via un réseau de transmission.

- + base de données et documentation centralisée
- + librairies de didacticiels expérimentés.

c) Implications au niveau des stratégies d'enseignement.

- + affinement des stratégies mises en évidence dans l'enseignement programmé
 - . stratégie linéaire (question-réponse)
 - elle n'a pas beaucoup changé, quant aux concepts utilisés, par rapport à la méthode de Skinner
 - elle fait appel essentiellement à la mémoire et a pour objectif l'acquisition et le contrôle de connaissances.
 - . stratégie ramifiée ou tutorielle
 - le principe d'adaptation, du chaînage des dialogues en fonction du type de réponse de l'élève reste à la base de cette stratégie, mais il peut être réalisé en tenant compte de renseignements plus riches: par exemple l'historique des dialogues et leurs résultats.
 - par contre, la nécessité du choix multiple de réponses (qui était due au support manuel de l'enseignement programmé) disparaît : le mode conversationnel d'utilisation de l'ordinateur permet l'acquisition de réponses ouvertes et leur analyse est réalisée automatiquement.
 - cette stratégie peut faire appel à des aptitudes plus diverses (déductions logiques, comparaisons, identifications, créativité, ...) et a un objectif plus pédagogique que la stratégie linéaire, dans la mesure où elle s'adapte au contexte des réponses de l'élève.
- + de nouvelles stratégies peuvent être envisagées :
 - . stratégie de résolution de problèmes
 - les solutions données par l'élève sont décomposées en un nombre d'étapes, successivement analysées et enregistrées. L'élève passe ainsi de réponses simples à des questions directes, à des constructions de plus en plus complexes.

- cette stratégie fait appel essentiellement à un esprit d'analyse, de synthèse, à la créativité. Elle a un objectif de recherche guidée.
- . stratégie de simulation, d'étude de cas
.....
- des cas réels sont simulés sur ordinateur, les réponses de l'élève sont discutées afin de mettre en évidence les différentes étapes d'une prise de décision
- cette stratégie fait appel à la créativité afin :
 - . de découvrir, à partir des expériences simulées, une relation entre les entrées et les sorties d'un modèle
 - . d'acquérir certaines expériences: l'élève donne les paramètres d'entrée, l'ordinateur calcule les paramètres de sortie
 - . de créer soi-même son modèle qu'on pourra tester sur des cas réels et comparer les résultats
 Elle a un objectif essentiellement de recherche plus ou moins libre.

d) Implications au niveau des différents principes de l'enseignement.

+ Structuration des connaissances.

Trois types de structuration marquant une évolution dans l'effort d'analyse, de l'unification, de modélisation de certaines connaissances :

- . la structuration des connaissances peut être inscrite dans la structure même de la rédaction du programme et ne peut dès lors être accessible qu' au moment de l'exécution même de ce programme.
- . la structuration des connaissances peut s'organiser sous forme standardisée dans une base de documentation, de façon formelle ou informelle, indépendante.
- . la structuration des connaissances peut se traduire par une arborescence de règles de décision, dans un système expert indépendant.

+ Principe de stimulation

La diversification des outils de présentation de l'information, un échange conversationnel entre l'apprenant et l'ordinateur, les différents modes d'intervention de l'apprenant, sont des moyens qui, dans une stratégie donnée, peuvent être dosés afin de stimuler l'élève et de le tenir en éveil.

+ Principe d'adaptation

Toutes les transactions échangées entre l'ordinateur et l'apprenant et leurs résultats, peuvent être mémorisées.

La décision de réorienter l'apprenant ou de lui faire sauter des parties de cours peut être prise en fonction d'indicateurs plus riches.

+ Principe de contrôle en permanence

Les procédures d'analyse et d'évaluation peuvent être plus précises et faire appel à des structures de connaissances plus complexes.

- + De nouveaux principes peuvent se dégager. Par exemple, un principe que l'on pourrait appeler le principe d'initiative : l'étudiant devient maître dans le dialogue, peut faire appel à des outils d'interrogation de documentations, de manipulation de données, à des modèles, ...

I .14. Evolution dans la conception des didacticiels :

- les objets pédagogiques.

a) Concepts

Parallèlement aux développements technologiques, une évolution dans la conception des logiciels s'est opérée :

- . dans un premier temps, les logiciels étaient conçus de façon à reproduire strictement la logique algorithmique des organigrammes
- . peu à peu, ils ont été réalisés en dégageant des fonctions orientées vers des tâches logiques bien déterminées, de façon à assurer une plus grande modularité dans la conception et la paramétrisation de ces fonctions
- . une autre forme de conception peut s'avérer plus appropriée à certains types de logiciels orientés vers la description de structures de données: l'approche en termes de types abstraits. Chaque type n'est accessible que par l'intermédiaire des fonctions opérant sur la structure de données associée.

La conception des logiciels d'enseignement tend à suivre cette évolution en proposant des outils informatiques reflétant les diverses approches :

- . des langages d'auteurs ont été développés afin de donner aux enseignants des primitives de base de mise en page des sorties sur écran, de fonctions graphiques, d'analyse de réponses, de suivi des performances;
- . Kenneth Tait ([19]) fait remarquer que ces langages inhibent plus qu'ils ne promeuvent une rédaction souple des didacticiels. Il propose une vue d'ensemble en termes de tâches logiques de plus haut niveau :
 - transmission de la matière enseignée
 - présentation d'écrans formatés
 - interaction, style d'échange entre l'ordinateur et l'élève
 - structuration, chaînage des actions, ...
- . Une nouvelle tendance fait apparaître une structuration des didacticiels en types de données, en objets pédagogiques

pouvant être manipulés séparément.

- Exemple du système IMAGO :

Les objets mis en évidence dans un didacticiel sont :

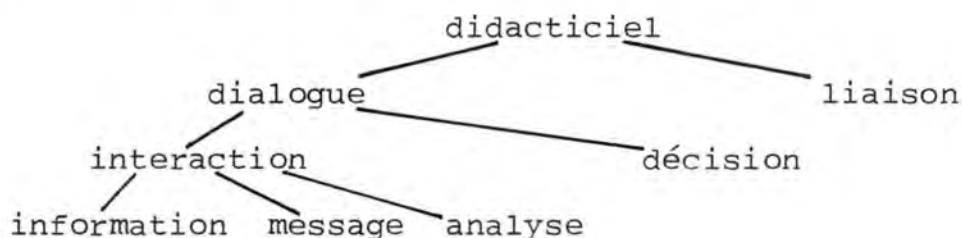
- ' des suites de blocs d'enseignement et de blocs de contrôle
 - ' un bloc d'enseignement représente un graphe orienté d'éléments de dialogue
 - ' chaque élément, noeud de dialogue, définit :
 - le texte de l'information adressée à l'élève
 - l'ensemble des expressions probables de l'élève auxquelles l'enseignant attribue une valeur sémantique
 - l'ensemble des commentaires associés à ces expressions.
 - ' chaque arc définit :
 - sa direction et les noeuds qu'il relie
 - les valeurs sémantiques qui le conditionnent
- Les objets qui tendent à être reconnus ([3]) sont :
- ' l'objet didacticiel: graphe des objets dialogue
 - ' l'objet dialogue: graphe des objets interaction élémentaire
 - ' l'objet liaison: règles de chaînage entre les objets dialogue dont les conditions font référence aux valeurs de variables mises à jour dans l'environnement dialogue
 - ' l'objet interaction: échange élémentaire entre l'ordinateur et l'élève
 - ' l'objet décision: règles de chaînage entre les objets échange dont les conditions font référence aux valeurs de variables mises à jour dans l'environnement échange.
 - ' l'objet information
 - ' l'objet message de l'apprenant
 - ' l'objet analyse de message de l'apprenant.

Ces objets sont hiérarchisés selon la relation "active" :

Exemple:

- en cours d'exécution, un objet didacticiel active un couple d'objets dialogue-liaison

- l'objet didacticiel constitue donc l'environnement d'exécution des objets dialogue et liaison :



b) Extensibilité

Cette conception modulaire des objets pédagogiques permet la création de nouveaux objets si le besoin s'en fait sentir. On peut penser à la création d'un objet environnement-message par exemple, intervenant au même titre que l'objet message dans la hiérarchisation des concepts.

Il décrirait les types de commandes que l'élève est autorisé à formuler pour investiguer l'environnement dans lequel il se situe :

- commandes concernant la consultation de différents types de documentations, structures de connaissance,
- commandes concernant l'utilisation d'outils ou modèles particuliers à exécuter sur base des paramètres introduits,
- commandes permettant à l'élève de désigner le dialogue ou l'interaction suivants (et donc de passer outre des décisions de chaînages prévus).

c) Paramétrisation

On peut étendre et affiner la variété de la paramétrisation de ces objets en fonction de la stratégie pédagogique que l'on veut appliquer et de la manière dont on veut mettre en oeuvre les différents principes pédagogiques.

Si l'on prend l'exemple de l'objet environnement-message qui vient d'être spécifié :

- il peut permettre une adaptation de stratégie :

- . la stratégie peut être de type recherche guidée si l'élève peut disposer :
 - ' d'un certain nombre de moyens d'investigation des connaissances et des modèles de l'environnement dans lequel il se trouve,
 - ' de commandes lui permettant d'orienter lui-même le dialogue vers d'autres parties de la leçon.
 - . La stratégie peut être de type acquisition de connaissances, et dans ce cas toutes les possibilités d'accès à d'autres documentations ou modèles, d'orientation du dialogue, sont masquées à l'élève.
- il permet un affinement des principes :
- . d'adaptation du dialogue par un masquage sélectif des possibilités offertes en fonction des difficultés, erreurs ou progrès de l'élève.
 - . de stimulation à la recherche, d'initiative.

I.2- Quelques applications de l'enseignement assisté par ordinateur dans le domaine de l'aide au diagnostic médical.

Présentons brièvement quelques exemples mettant en évidence certains des types de structuration des connaissances et certaines stratégies parmi ceux cités plus haut. Les connaissances et modèles mis en oeuvre sont communs à ceux des systèmes Aide au Diagnostic Médical associés :

a) Système d'aide au diagnostic thérapeutique de la digitaline ([25])

- structuration des connaissances :

celles-ci sont incorporées dans l'algorithme de forme arborescente du programme du système.

- stratégie :

- . le système est orienté essentiellement vers la simulation, la recherche
- . il autorise des retours arrière pour modifier la valeur de certains paramètres et recalcule la valeur des paramètres dépendants
- . il est capable d'expliquer le chemin parcouru en passant en revue les procédures appelées, l'utilité de certaines variables en relevant les relations d'interdépendance entre variables, ...

- remarque :

cette représentation procédurale des connaissances peut permettre une modélisation très fine. Les systèmes engendrés sont cependant très limités, rigides et spécifiques à un domaine fort particulier.

b) Système d'aide à l'enseignement de l'hématologie ([4])

- structuration des connaissances :

- . pour l'enseignement de la cytologie, des documents sont conservés sur diapositives; ces connaissances sont donc informelles
- . en ce qui concerne l'enseignement du diagnostic des anémies, les différents diagnostics possibles et les examens complémentaires associés ont été structurés par contextes constituant une arborescence.

- stratégie :

- . en cytologie, la stratégie est tutorielle et a pour objectif la reconnaissance et l'analyse d'images visualisées sur diapositives
- . en ce qui concerne l'aide au diagnostic, la stratégie adoptée
- . est de type simulation guidée et a pour objectif de forger le raisonnement, la réflexion.

- remarque :

ce système reste très spécialisé et ne permet pas, en tant que tel, une interrogation indépendante des connaissances; il reste donc assez rigide.

c) Un système d'aide à l'enseignement du diagnostic médical développé à la Faculté de médecine de Rennes ([6])

- structuration des connaissances :

- . elles sont structurées de façon standard dans une base de documentation et concerne la description générale de plus de 2000 maladies (fréquence qualitative de la maladie, signes fréquents, spécifiques, pronostic, données complémentaires, biologie, maladies associées, ...)

Le système d'information associé permet une recherche documentaire (description d'une maladie, liste des maladies contenant un symptôme, ...)

- . une recherche des maladies les plus pertinentes sur base d'un groupe de symptômes obligatoires et facultatifs
- . une recherche des maladies les plus probables en favorisant le choix d'examens complémentaires.

- stratégie :

la stratégie est tutorielle et a pour objectif la réflexion, l'acquisition d'un esprit de déduction.

Le système autorise un recours aux fonctions de documentation.

- remarque :

la structuration des connaissances en système d'information permet une recherche indépendante, comparative, qui en favorise l'étude, l'analyse, la synthèse.

La stratégie adoptée reste cependant assez rigide quant aux possibilités de création d'unités d'enseignement.

d) Le système GUIDON associé au système expert MYCIN :

- structuration des connaissances :

. elles sont représentées par un ensemble :

- de faits : données passives représentant la déclaration des objets de base
- de règles : de production, d'inférence, sous forme de prédicats du premier ordre, exprimant les relations entre ces faits.

L'ordre de ces faits et règles n'ayant, en principe, pas d'importance, la connaissance est structurée en une arborescence ET/OU.

. elle est constituée également de métarègles permettant de superposer une organisation particulière à celle de l'arborescence de base.

. le mode d'interrogation de cette base de connaissances est identique au mode de représentation de ces connaissances.

- stratégie :

ce type de connaissance autorise une stratégie de recherche non limitée.

- remarque :

ce mode de représentation des connaissances est le plus souple, tant au niveau de l'interrogation qu'au niveau de l'extensibilité de la base et permet une réelle recherche, mais il ne peut s'appliquer que dans des domaines bien délimités et spécifiques.

I.3- Rédaction des didacticiels.

a) Nécessité d'un travail d'équipe.

Afin qu'un logiciel d'enseignement puisse être à la fois efficace, fiable, portable, d'une conception modulaire, qu'il fasse apparaître des structures de connaissances éprouvées selon des techniques, principes et stratégies de l'enseignement expérimental, il devient nécessaire que les enseignants travaillent en collaboration entre eux et avec des psychopédagogues, des cybernéticiens (lorsque l'on aborde les problèmes de l'intelligence artificielle) et des informaticiens.

b) Les différentes phases de la rédaction.

1. Préliminaires

- il faut s'assurer que la matière est programmable sans trop de difficultés, avec le moins d'imprécision ou ambiguïté possible, les classes de valeurs de réponses possibles devant pouvoir être définies avec précision
- pour que le programme soit efficace, il faut s'assurer du degré de motivation des étudiants
- il est nécessaire de déterminer et définir la place de ce mode d'enseignement dans la formation de l'étudiant

2. Détermination des objectifs en termes de comportements terminaux.-----

- avant tout, il faut déterminer précisément les connaissances prérequis,
- ensuite, les objectifs seront déterminés en termes de performances et de comportements terminaux observables et mesurables à atteindre, et qui permettront d'évaluer précisément les progrès effectués.

Ces comportements ou aptitudes peuvent être de type :

- . association
- . différenciation
- . discrimination multiple ou identification
- . chaîne précise de comportements
- . concept de classes d'objets
- . enchaînement de concepts ou principes

- . enchaînement de concepts avec concepts intermédiaires ou stratégie.

3. Structuration de la matière.

Il existe deux types de structurations sous-jacents à cette structuration de la matière :

+ structuration des connaissances :

déterminer la nature des concepts, leurs liens: cette structure est intrinsèque à la matière enseignée.

Voici quelques méthodes de structuration des connaissances locales d'une matière :

- raffinement successif des concepts

Le comportement terminal est synthétisé en une phrase aussi précise que possible; chacun de ses termes est repris et explicité à son tour en une phrase. Ce processus est appliqué de façon récursive.

- analyse de la matière en règles élémentaires

Après construction de la matrice exprimant toutes les relations entre ces règles élémentaires, on peut voir apparaître plus ou moins précisément, des zones figurant les principaux concepts (interrelations fortes entre règles élémentaires)

- construction d'une démarche appropriée à la matière envisagée, dans le cas où des méthodes universelles ne fonctionnent pas.

Il s'agit alors de déterminer les plus petites entités que toutes les personnes exerçant la discipline enseignée désignent d'un commun accord comme élémentaires, ainsi que les opérateurs agissant sur ces entités, liés à la sémantique de ces entités.

- analyse à rebours

- . on passe en revue toutes les propriétés du concept envisagé en imaginant le plus grand nombre de situations possibles dans lesquelles ce concept intervient:

on peut ainsi énoncer un ensemble de propositions au sujet de ce concept.

- . de ces propositions, on tente d'extraire tous les concepts de base, une proposition étant une combinaison de ces "atomes". Soit E l'ensemble de ces atomes.
- . on construit ensuite l'ensemble des parties de E et la matrice ou treillis de Boole associé (treillis conceptuel) : ensemble des concepts atomiques.
- . à partir des concepts atomiques, on construit toutes les propositions formulables: on obtient le treillis propositionnel.

Cette démarche a été adoptée pour la structuration des informations du système IMAGO ([21])

+ structuration de la présentation des connaissances :

déterminer l'ordre de présentation des différents concepts: c'est la structure "algorithmique" des notions enseignées par laquelle s'exprime la stratégie que l'on veut adopter et les principes à mettre en oeuvre pour atteindre les objectifs :

- la matière étant structurée, il s'agit de déterminer l'ensemble des valeurs des réponses que l'étudiant est susceptible de proposer et de leur attribuer une valeur sémantique. Ensuite, en appliquant le principe d'adaptation, en fonction de la stratégie adoptée, on détermine la structure algorithmique des séquences d'interactions.
- le mode de présentation de la mise en forme des informations, des connaissances a une importance pédagogique : sous forme graphique, sous forme formelle ou imagée, il fait appel à des aptitudes différentes: abstraction, imagination.

4. Rédaction proprement dite

Les outils informatiques proposés pour la rédaction des didacticiels ont suivi l'évolution de la conception de ces logiciels (cfr. supra IIe partie I.1- 4) :

- il existe plusieurs langages d'auteurs offrant des facilités de mise en page des sorties sur écran, d'analyses des réponses, de fonctions graphiques, du suivi des performances des étudiants, ...
(Exemple: TUTOR de PLATO, PILOT, ...)
- certains outils proposent un ensemble de fonctions de plus haut niveau orientées vers des tâches pédagogiques (exemple: le système GALTS: Generated Author Language Teaching System) ([19])
- pour prendre en charge la construction des objets pédagogiques, des éditeurs spéciaux ont été créés.

5. Mise au point - validation des didacticiels.

En vue d'améliorer la qualité du didacticiel au point de vue éducatif, il est nécessaire :

- d'effectuer un contrôle des traces des différents dialogues réalisés entre l'ordinateur et un ensemble d'étudiants lors de phases d'expérimentation du logiciel, afin de mesurer les taux de réponses erronées, de mettre en évidence les types d'erreurs commises, les points de blocage, les situations non prévues qu'il faudra réanalyser, ...
- d'effectuer une analyse au niveau de la motivation des étudiants pour certains types d'exercices, de l'adaptation réelle du système à leurs demandes et à leurs difficultés, ... ce qui permettrait de restructurer la matière et de trouver une meilleure organisation de la formation globale du médecin en utilisant d'autres moyens (techniques ou non) d'enseignement.

II. SYSTEME D'AIDE A L'ENSEIGNEMENT DU DIAGNOSTIC HOMEOPATHIQUE

II. 1- PRESENTATION DU PROJET.

II-11. Objectifs :

Dans le cadre de la formation générale du médecin étudiant en homéopathie, certains domaines ont été considérés comme pouvant être pris en charge et mieux exercés par un système automatisé d'aide à l'enseignement.

Essentiellement :

- l'initiation à la technique répertoriale :

- Apprendre - à interpréter un symptôme observé
 - à trouver le mot précis et visualiser son emplacement dans le répertoire afin de déjouer les pièges classiques de l'interrogation et de la répertorisation.

- la pratique du diagnostic et de la thérapeutique homéopathique

Confronter le médecin à des cas concrets de malades guéris en principe par l'homéopathie, en permettant un accès aux différents outils de consultation et différentes méthodes existants, en mettant en évidence les lois, principes et règles de la démarche homéopathique, tant au niveau du diagnostic homéopathique (hiérarchisation, sélection des symptômes, ...) qu'au niveau du diagnostic remédial (diagnostic différentiel, méthode de sélection..

Mais d'autres aspects de la démarche pourraient être envisagés :

- l'analyse de l'évolution d'une maladie (loi de Hering, pronostic,...)
- second diagnostic remédial
(cfr. Première partie I.4-)

II-12. Caractéristiques pédagogiques

II-121. Stratégies et aptitudes à mettre en évidence

+ Initiation à la technique répertoriale

- . la stratégie tutorielle semble la plus adaptée afin de réajuster les conclusions émises par l'étudiant

- . elle a pour objectif l'acquisition de connaissances, d'un esprit de discrimination.
- + Pratique du diagnostic et de la thérapeutique homéopathique
 - . une stratégie de recherche en permettant une modulation de l'initiative de l'étudiant par le biais de la création d'un environnement donnant accès aux documentations, modèles, primitives existants, convient mieux pour ce type d'enseignement.
 - . elle a pour objectif l'acquisition d'un esprit d'analyse, de comparaison, de déduction, de synthèse, de recherche.

II-122. Types de structures de connaissance utilisés

- + Les connaissances associées à l'initiation de la technique répertoriale sont essentiellement locales dans le contexte sémantique précis, déterminé par une information informelle (texte extrait d'une Matière Médicale, d'un traité de toxicologie, ou texte descriptif des examens d'un patient réel ou fictif). Elles seront donc décrites dans le cadre d'un dialogue de façon exhaustive; elles peuvent être le résultat d'applications de primitives sur des connaissances globales (pathogénésie d'un remède, symptômes déterminés par mots-clés, ...) Elles seront décrites par la référence à la primitive utilisée et les paramètres associés.
- + Les connaissances associées à la pratique du diagnostic et de la thérapeutique homéopathique peuvent, elles, être extraites de modèles globaux de connaissances (résultats de méthodes de sélection de remèdes sur base d'un ensemble de symptômes, de statistiques, ... cfr. supra Première partie III A.D.M.) . Elles seront donc décrites par la référence à la méthode utilisée et les paramètres associés.

II-13. Le poste de travail.

- + Un micro-ordinateur ou un terminal relié à l'ordinateur central
- + Ce poste est constitué d'un clavier alphanumérique et d'un écran permettant certaines possibilités graphiques.

II-14. Conception du système.

+ Démarche adoptée : objets pédagogiques.

Du fait de la diversité des sources d'informations, des modes d'acquisition et d'analyse de réponses, des types de connaissances et des méthodes, l'approche en termes d'objets pédagogiques a semblé être mieux appropriée pour assurer la souplesse et la modularité des composants d'une leçon et permettre ainsi :

- . une paramétrisation plus précise, spécifique à chaque objet
- . une extensibilité plus facile en cas de mise en évidence
 - de nouveaux objets
 - de nouveaux paramètres
- . la création de primitives de gestion strictement attachées à chacun des objets.

+ Spécification des objets pédagogiques retenus dans le projet

1) Information transmise

L'information pouvant être transmise se trouve sur fichier.

- Actuellement :

- . elle est locale au système d'aide à l'enseignement, soit au niveau global du système, soit au niveau d'une leçon, d'un dialogue, d'une interaction, d'une décision ou d'une liaison particulière.
- . elle est de type texte
- . elle peut utiliser certaines primitives de présentation :
 - affichage en mode souligné, gras, clignotant, inversé, normal
 - positionnement à une colonne particulière
 - saut d'une ou plusieurs lignes
- . elle peut se présenter sous certains formats particuliers :
 - lorsqu'une information est destinée à être utilisée pour présenter un choix multiple de réponses, celles-ci doivent être précédées d'une commande demandant sa numérotation
 - lorsqu'une information comporte des zones destinées à être remplies ultérieurement, celles-ci doivent être précédées d'une commande déterminant le début de la zone

et le type de valeur attendue (symptôme, remède, la maladie, nombre réel, entier, chaîne de caractères).

- Extensions :

- . elle pourra être globale :
 - extraite des Matières Médicales par mots-clés
 - comme résultat de modèles de connaissance (statistiques, schémas diagnostiques, ...)
 - présentant le dossier d'un patient réel ou fictif extrait de la base des cas
- . elle pourra être de type tabulé ou graphique (en fonction des primitives disponibles).

2) Message de l'étudiant

.....

Le périphérique d'entrée des données dans l'ordinateur étant un clavier alphanumérique, les modes d'acquisition de message retenus sont :

- le marquage de mots ou suites de mots dans un texte :
Ce mode permettra, par le biais de commandes, de mouvoir le curseur et de délimiter les zones à sélectionner.
La liste des zones sélectionnées pourra être extraite du texte et affichée en vue de modifications ou suppressions.
- le remplissage de zones libres dans un texte :
ce mode permettra de sélectionner la zone à remplir au moyen de commandes déplaçant le curseur.
Le type de valeurs à acquérir dans chacune des zones étant défini à la conception du texte, un test syntaxique pourra être réalisé dès la saisie du message.
- la présentation d'un choix de réponses :
ce mode permettra l'acquisition d'un nombre ou suite de nombres déterminant les réponses retenues.
- la forme ouverte :
ce mode permettra d'entrer une liste de valeurs d'un format particulier. Le format du message peut être de type :
 - . libre
 - . structuré en un ou plusieurs champs, chaque champ étant

déterminé par :

- le type élémentaire des valeurs admises pour ce champ (symptôme, remède, maladie, entier, nombre réel, chaîne de caractères).
- la longueur maximum des valeurs acceptables
- un indice nécessaire afin de déterminer l'ordre dans lequel seront évalués les différents champs.

3) Modèle de connaissance

- Un objet modèle de connaissance structure l'ensemble des objets connaissances locaux à un contexte particulier, chacun de ces objets étant associé à une classe de valeurs d'un format déterminé.
- Actuellement, la seule relation logique unissant ces objets est inscrite dans le fait qu'ils décrivent une même situation pédagogique. D'autres relations, de type arborescence par exemple, pourraient éventuellement être envisagées si une telle structure est capable de refléter une situation déterminée.

4) Connaissance

Un objet connaissance est associé à un type de format de message déterminé. Il décrit un ensemble de valeurs de référence.

La connaissance peut être de type :

- local :

- . la connaissance de référence est dépendante du contexte particulier, non formalisable de la leçon, du dialogue ou de l'échange qui l'utilise.
- . les différentes valeurs envisageables du type du format associé sont définies de façon exhaustives, plusieurs valeurs pouvant être déclarées synonymes.
- . la seule organisation des valeurs qui a pu être mise en évidence actuellement est une classification par degré de pertinence, dans le contexte qui l'utilise. Les

degrés de pertinence retenus sont : correct et obligatoire, tolérable et non obligatoire, refusé.

- . à chaque classe de valeurs peut être associé un commentaire.

- global :

- . la connaissance de référence est inscrite dans un modèle ou une primitive externe, global, indépendant du contexte qui l'utilise.
- . dans le cadre actuel du système, les modèles disponibles sont :
 - dans le cas où le format associé à la connaissance est de type remède :
 - ' modèle des fréquences
 - ' modèle des degrés
 - ' modèle ELECTRE II
 - dans le cas où le format associé à la connaissance est de type symptôme :
 - ' diagnostic différentiel de deux ou trois remèdes (cfr. Première partie III A.D.M.)
- . les paramètres d'entrée sont extraits du modèle de connaissance auquel l'objet connaissance appartient.
Le modèle de connaissance doit donc contenir au moins un autre objet connaissance dont le format associé est identique à celui des paramètres d'entrée du modèle.
- . les valeurs issues du modèle peuvent être classées en intervalles disjoints auxquels sont attribués un degré de pertinence.

5) Modèle d'analyse
.....

Un modèle d'analyse est associé statiquement à un format particulier; il sera dynamiquement associé à un objet modèle de connaissance ou à un objet connaissance lors de son activation.

Il décrit une séquence de commandes qui devront être exécutées lors de l'analyse du message.

Ces commandes sont de type :

- activation d'un modèle :
 - . ayant pour paramètres d'entrée: les messages de l'apprenant dont le format est identique au format de ces paramètres
 - . ayant pour paramètres de sortie: une valeur dont le format est celui du message à analyser.
- activation de primitives de comparaisons
 - . ayant pour premier terme les messages de l'apprenant dont le format correspond à celui du message à analyser
 - . ayant pour second terme:
 - soit les messages corrects de même format extraits du modèle de connaissance qui sera associé dynamiquement au modèle d'analyse,
 - soit les messages de même format ayant été générés par l'activation d'un modèle.

Il existe deux primitives de comparaisons standards :

- . comparaison nuancée en fonction des degrés de pertinence des classes de valeurs de l'objet connaissance associé dans le cas où celui-ci est de type local
- . comparaison différentielle qui recherche les parties propres et l'intersection de deux ensembles de valeurs

Ces primitives ont pour résultat :

- . d'attribuer un degré de pertinence à chacune des réponses de l'apprenant en fonction du contenu de l'objet connaissance associé,
- . de déterminer les réponses manquantes
- . de relever les réponses qui n'ont trouvé aucun correspondant lors de l'analyse.

- activation de primitives d'évaluation standard :

Deux primitives d'évaluation sont prévues actuellement :

- . calcul de la moyenne pondérée par un facteur de pertinence, de la comptabilité des types de réponses relativement au nombre de réponses correctes à fournir,
- . dans le cas de réponses à choix multiple, il est possible de demander à l'apprenant de donner un indice de certitude à sa réponse; il sera pris en compte dans le calcul d'évaluation.

D'autres primitives de ces trois types, ou d'autres types de primitives, peuvent être envisagées si l'usage en démontre la nécessité.

6) Echange (ou interaction)

Un objet échange décrit la séquence des commandes définissant une interaction élémentaire entre l'apprenant et l'ordinateur opérant sur les objets de base: information, message, modèle de connaissance, modèle d'analyse.

Ces commandes sont de type :

- activation de la transmission d'une information
- activation de l'acquisition des messages de l'apprenant avec :
 - . la spécification du modèle de connaissance associé au contexte de l'échange,
 - . la spécification des formats de messages, référenciés dans le modèle de connaissance, que l'étudiant est autorisé à utiliser; ceux-ci seront donnés dans l'ordre dans lequel ils seront analysés,
 - . la spécification des types de consultation et des types de requêtes autorisées.
- activation des modèles d'analyse avec la spécification du modèle d'analyse associé à chacun des messages dont le format a été autorisé.

Si cela s'avère nécessaire, un nouvel objet pourrait être créé : l'objet environnement-message, qui reprendrait la spécification des types de consultation et de requêtes autorisées à laquelle serait ajoutée la spécification de nouvelles possibilités de commandes, afin de pouvoir moduler plus finement l'initiative accordée à l'étudiant.

7) Décision

Un objet décision décrit les règles de chaînage entre des objets échanges.

- Une condition de chaînage est représentée par une conjonction d'expressions relationnelles simples du type :

$$\langle \text{variable} \rangle \quad \left\{ \begin{array}{c} = \\ > \\ < \end{array} \right\} \quad \left\{ \begin{array}{c} \langle \text{valeur} \rangle \\ \langle \text{variable} \rangle \end{array} \right\}$$

où variable

identifie un objet de l'environnement de l'échange qui vient de se terminer, ou du dialogue courant, dont la valeur a été attribuée au cours de l'exécution de cet échange ou dialogue :

- lors de l'activation de l'échange :
l'identifiant de l'échange est ajouté à la liste des échanges déjà exécutés, ainsi que le nombre de fois que cet échange a déjà été exécuté, dans l'environnement-dialogue,
- lors de l'acquisition des messages :
les formats, les consultations, les requêtes qui ont été utilisés, ainsi que le nombre de fois que ces objets ont été utilisés, dans l'environnement-échange,
- lors de l'analyse des messages :
le nombre des messages classés selon les différents degrés de pertinence, l'évaluation globale, le temps de réponse, dans l'environnement-échange.

où valeur

est une constante du type de la variable du premier membre de la relation.

- Le second membre de la règle décrit une séquence de commandes exprimant la réaction engendrée par la vérification des conditions :
 - activation de l'affichage :
 - . de la valeur d'objets de l'environnement-échange ou dialogue
 - . d'information
 - . de commentaires engendrés lors de l'évaluation des messages de l'échange qui vient de se terminer
 - . de l'ensemble de l'évaluation des messages
 - . de la réponse correcte
 - activation d'un nouvel échange
 - répétition de l'échange qui vient de se terminer
 - continuer avec un échange déterminé, ainsi l'environnement de l'échange qui vient de se terminer n'est pas détruit
 - activation d'une nouvelle décision
 - arrêt du dialogue.

Les types d'objets dans les différents environnements et les commandes proposées au niveau du second membre des règles de chaînage ne sont pas exhaustifs.

De la richesse des différents environnements dépend la souplesse du chaînage.

8) Dialogue

Un objet dialogue spécifie l'environnement dans lequel devra se dérouler l'enchaînement des différents échanges pour atteindre son objectif, et plus précisément la première décision qui amorcera le dialogue.

On peut envisager qu'un dialogue puisse être paramétré afin de pouvoir l'exécuter sur plusieurs jeux de valeurs distinctes.

9) Liaison

Un objet liaison décrit les règles de chaînage entre les différents dialogues.

- Une condition de chaînage est représentée par une conjonction d'expressions relationnelles simples du type :

$$\langle \text{variable} \rangle \quad \left\{ \begin{array}{l} = \\ > \\ < \end{array} \right\} \quad \left\{ \begin{array}{l} \langle \text{variable} \rangle \\ \langle \text{valeur} \rangle \end{array} \right\}$$

où variable

identifie un objet de l'environnement du dialogue qui vient de se terminer ou de la leçon courante, dont la valeur a été attribuée au cours de l'exécution de ce dialogue ou de cette leçon :

- lors de l'activation du dialogue :
l'identifiant du dialogue est ajouté à la liste des dialogues déjà exécutés, ainsi que le nombre de fois que ce dialogue est exécuté, dans l'environnement-leçon,
- lors de la terminaison du dialogue :
évaluation globale, temps de réponse, dans l'environnement dialogue.

où valeur

est une constante du type de la variable du premier membre de la relation.

- Le second membre de la règle de chaînage décrit une séquence de commandes qui devront être exécutées en cas de vérification des conditions :
 - activation de l'affichage :
 - . d'informations
 - . de l'évaluation du dialogue
 - activation d'un nouveau dialogue

- répéter le dialogue
- activer une nouvelle liaison
- arrêt de la leçon.

Les types d'objets dans les différents environnements et les commandes proposées au niveau de la réaction ne sont pas exhaustifs.

10) Leçon

Un objet leçon spécifie l'environnement dans lequel devra se dérouler l'enchaînement des différents dialogues, et plus précisément la première liaison qui amorcera la leçon.

II.2 - ANALYSE FONCTIONNELLE

21 - DESCRIPTION ET SPECIFICATION DES DONNEES

21.1 - Description des données

21.11 - méthode

- le système d'information associé au projet vu en termes d'objets pédagogiques fait apparaître nettement des entités de base. Le modèle entité-association de description et spécification de données semble approprié pour mettre en évidence clairement la structure et les attributs de ces objets (méthodologie ([5] p. 14)).

Dans une première phase, les objets seront donc représentés et décrits dans ce modèle.

- les relations entre les objets de base n'ayant aucune propriété, le modèle entité-association est trop lourd pour les exprimer. Dans une seconde phase, les relations entre les objets seront exprimées dans un modèle relationnel.

21.12 - description des objets dans le modèle E/A.

description des objets :

.....

- . main
- . cas
- . leçon
- . dialogue
- . liaison
- . échange
- . décision
- . info
- . format_mess
- . modèle_analyse
- . mod_conn
- . connaissance.
- . trace

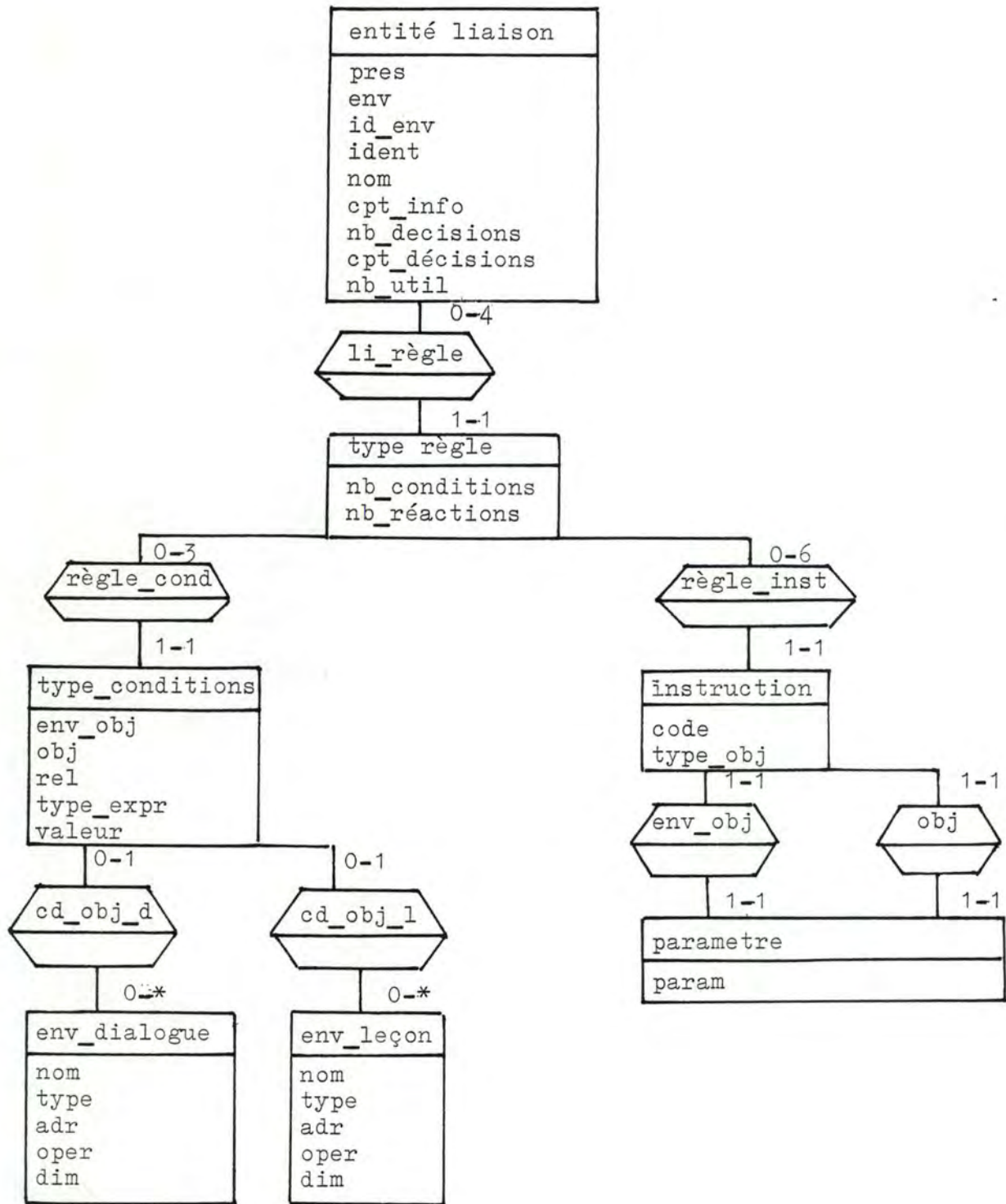
Rem: dans les schémas qui suivent, l'indication [*] signifie que l'entité correspondante a été décrite auparavant.

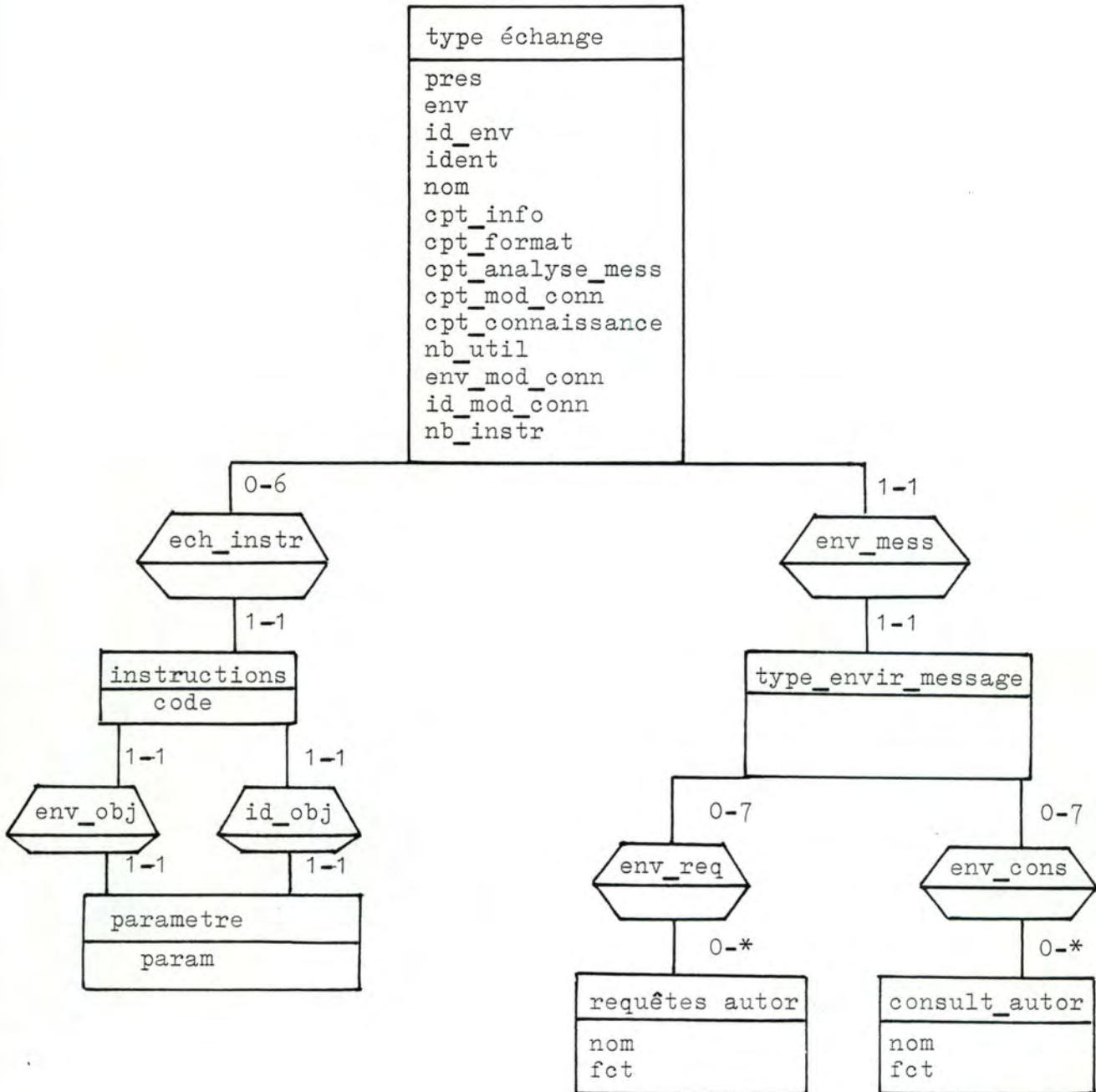
entité main
cpt_leçon
cpt_dialogue
cpt_cas
cpt_format
cpt_info

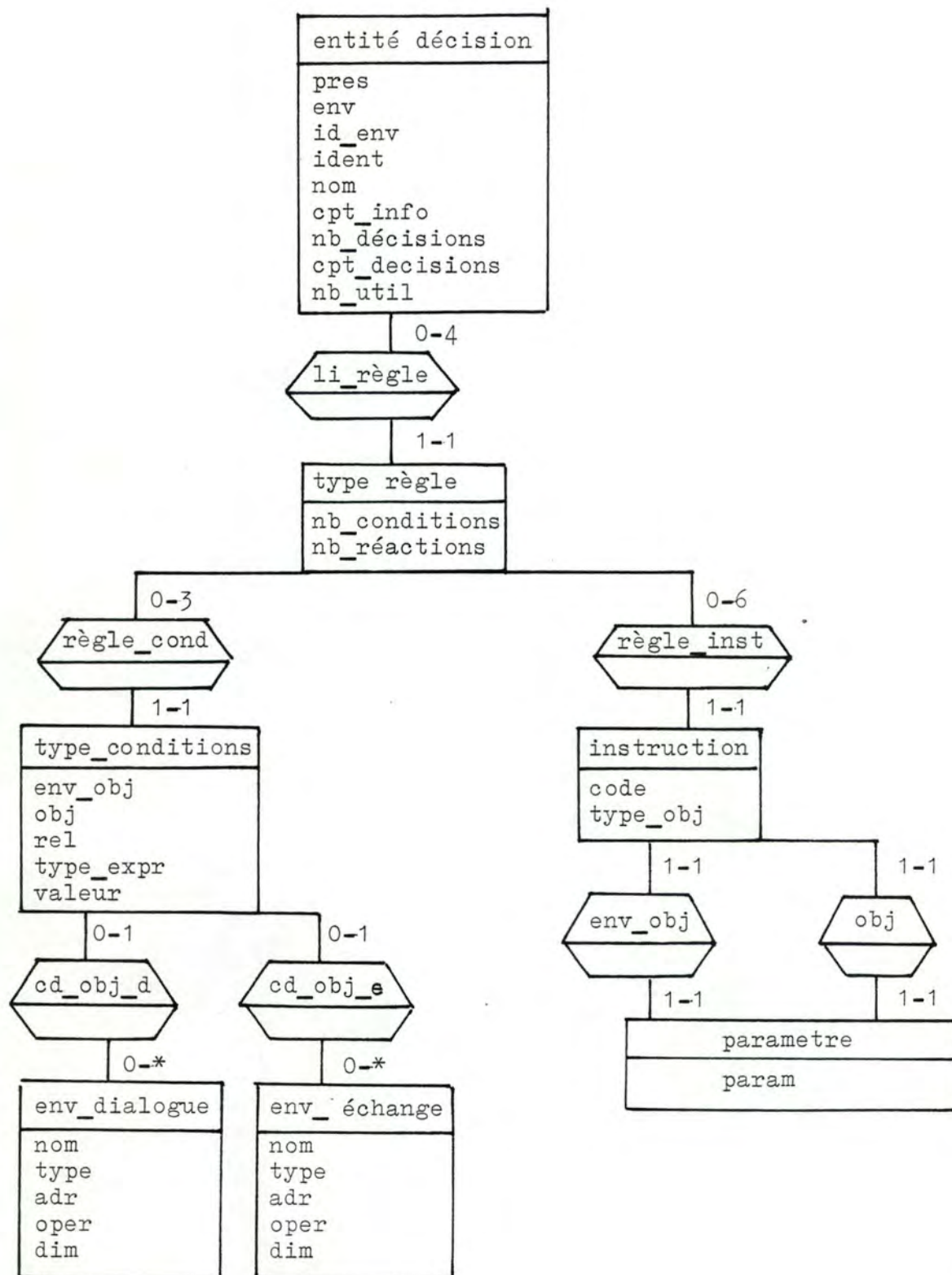
entité cas
pres
ident
nom
objectif
id_doc
no_dos
type

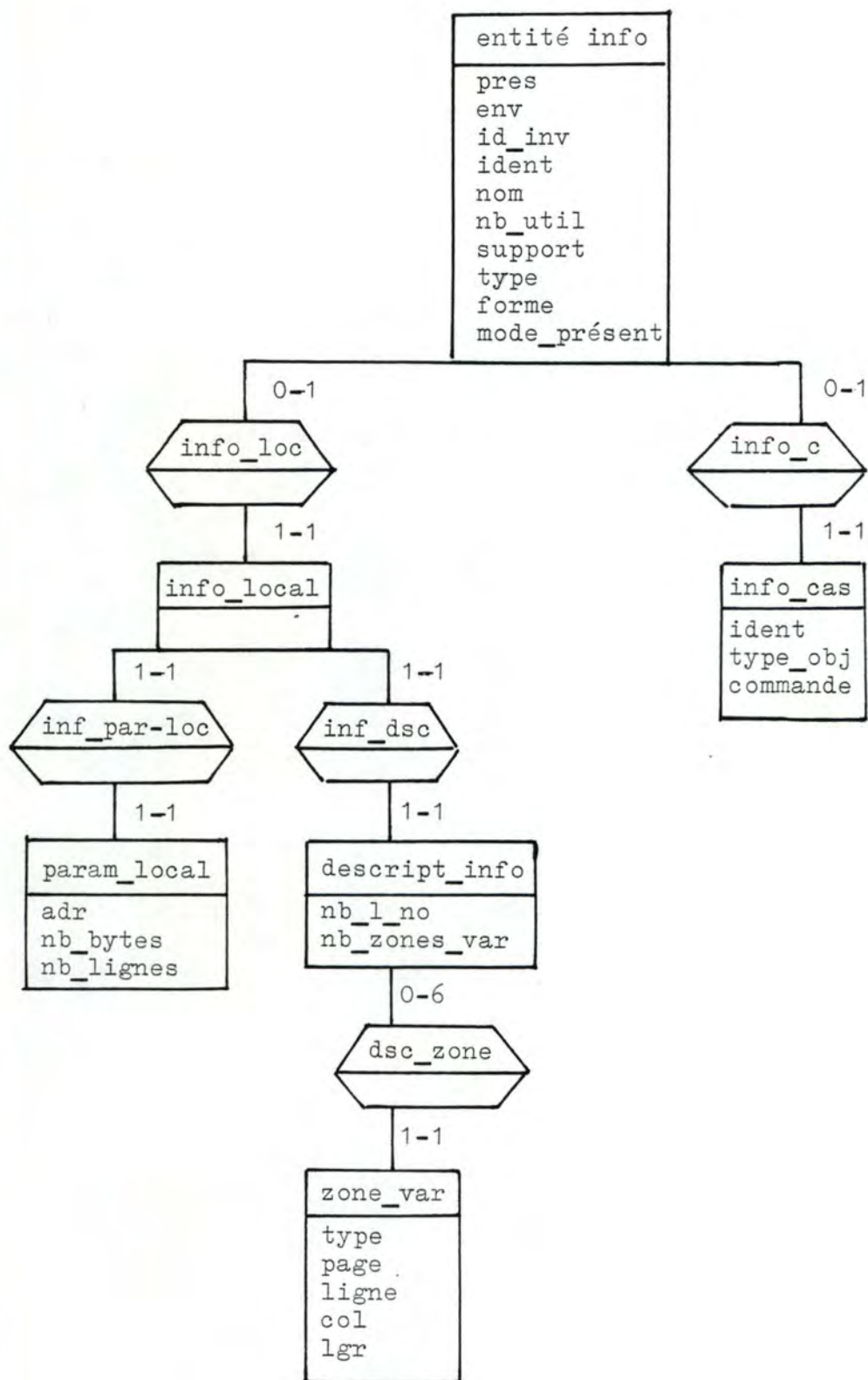
entité leçon
pres
ident
nom
objectif
difficulté
cas
optim
cpt_dialogue
cpt_liaison
cpt_info
id_liaison_intro

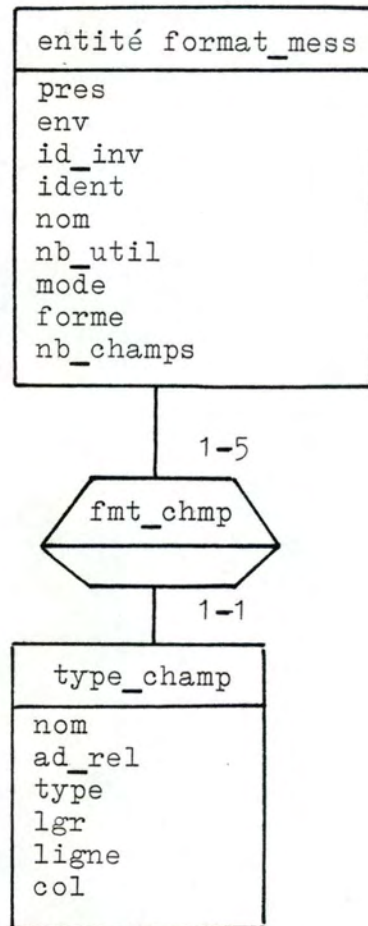
entité dialogue
pres
env
id_env
ident
nom
objectif
difficulté
cas
cpt_échange
cpt_décision
cpt_info
cpt_format
cpt_analyse_mess
cpt_mod_conn
cpt_connaissance
optim
nb_util
id_décision_intro

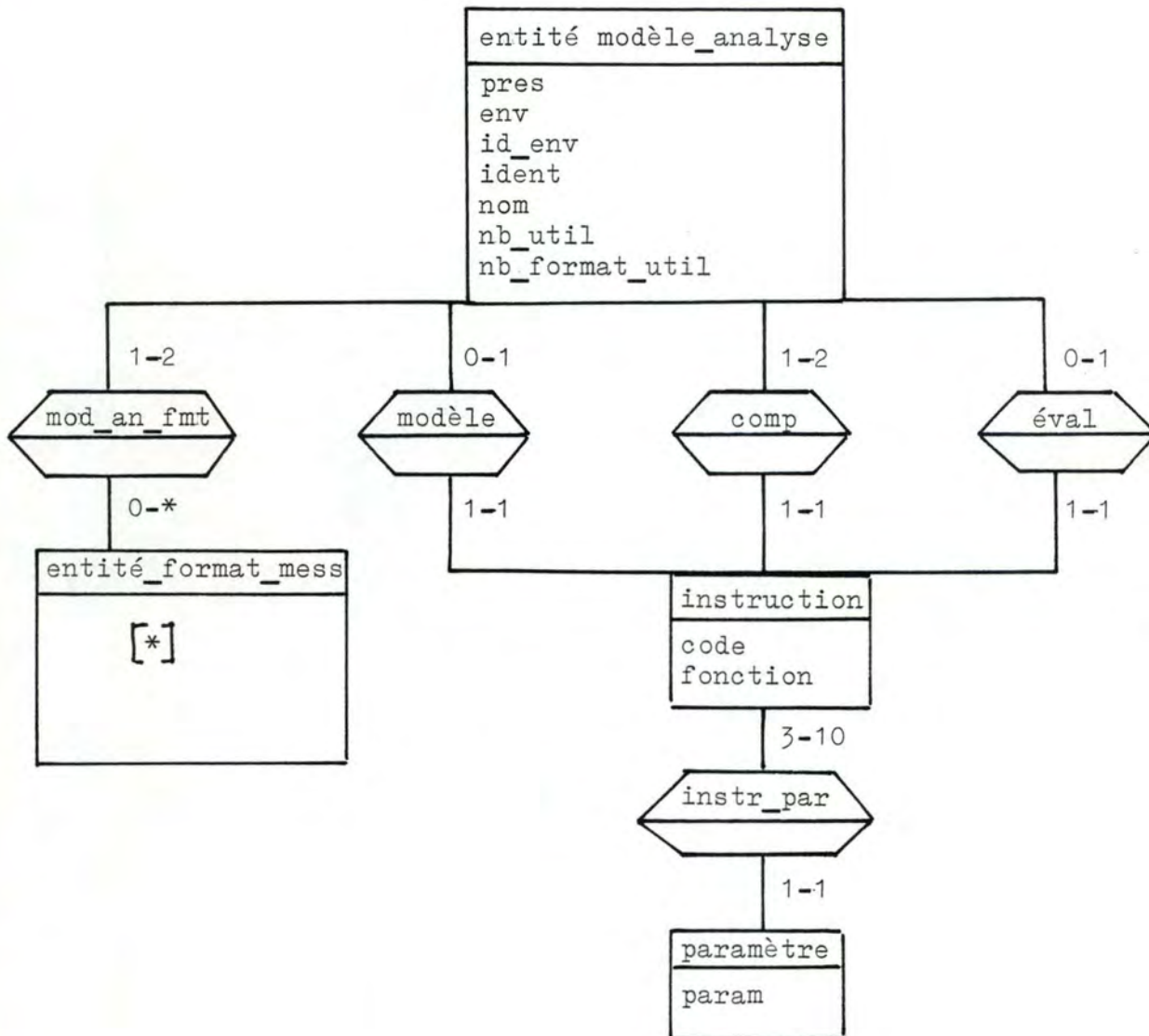


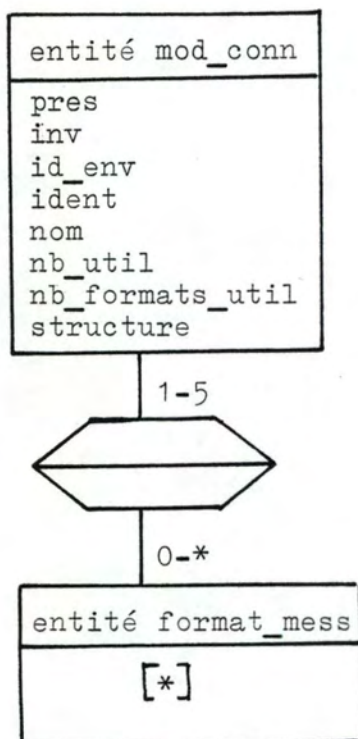


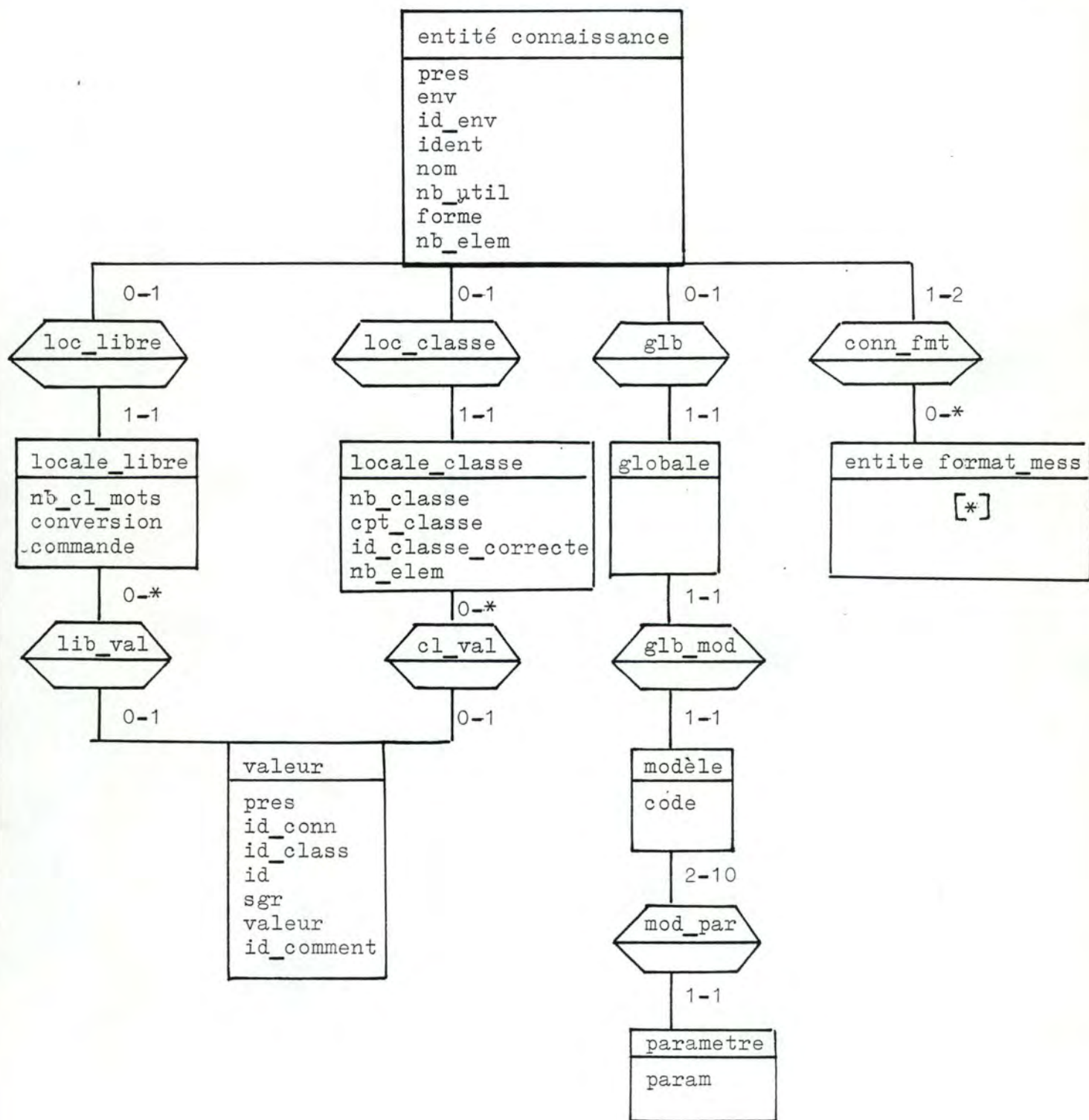


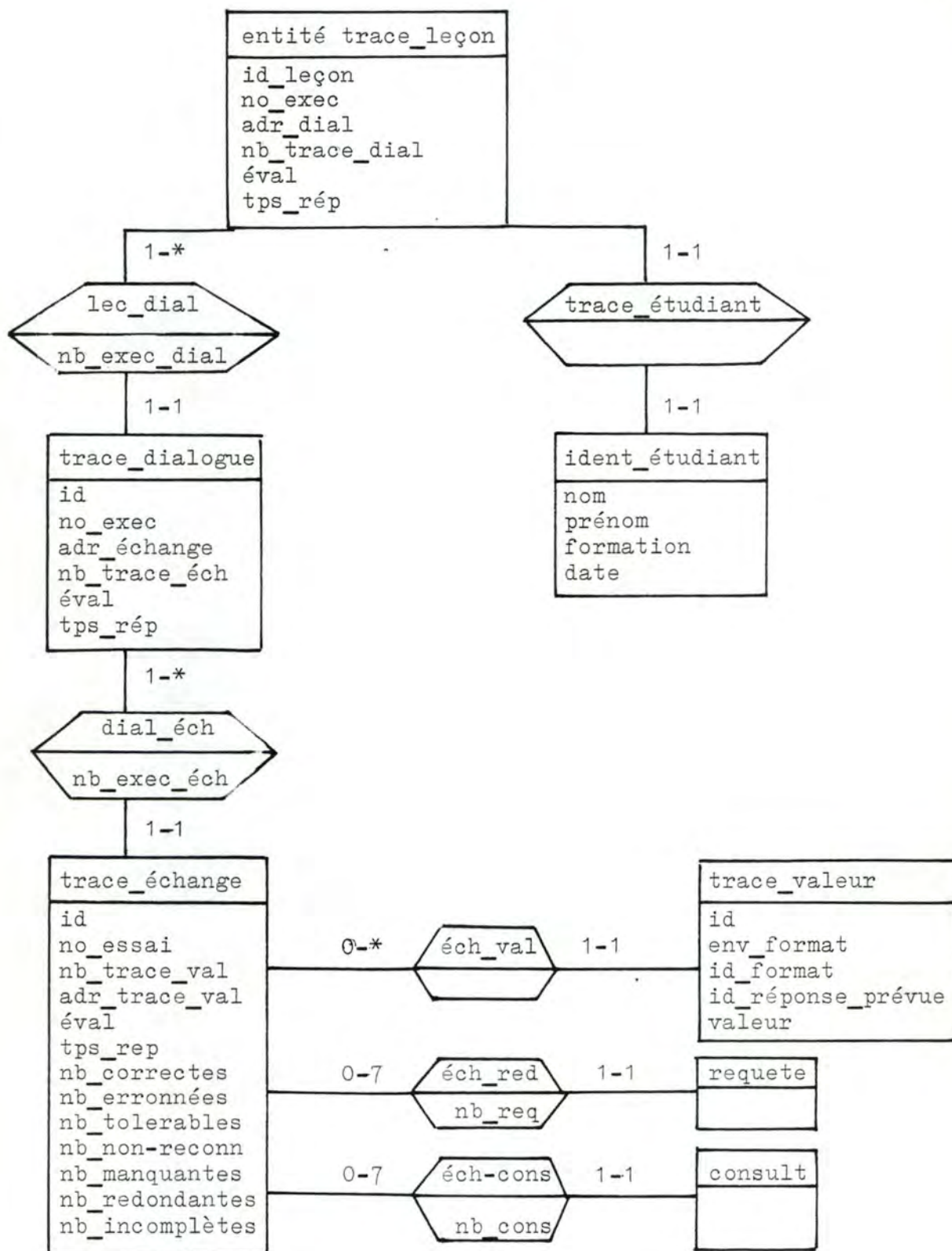






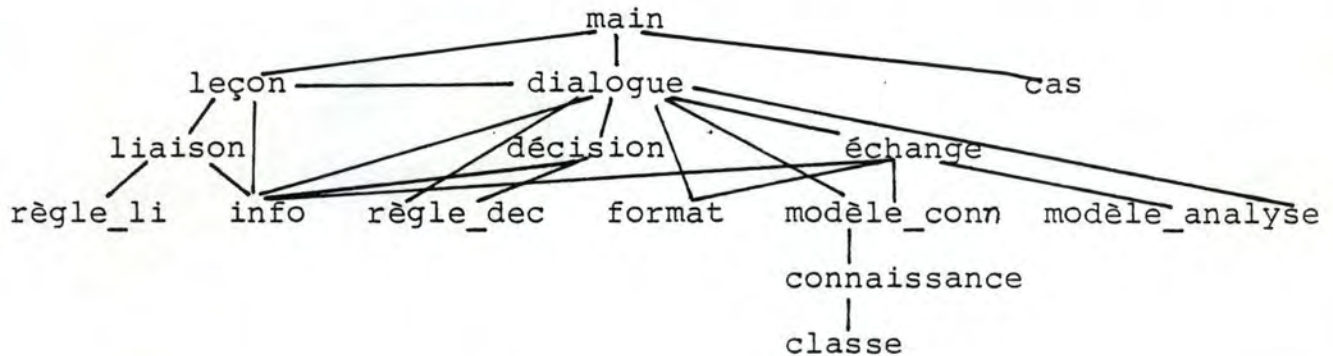






21.13 - description des relations entre les objets

- . graphe actuel de la relation "valide-est validé dans"

remarques

- ce graphe est supporté par la structure auxiliaire "environnt" (cfr ANNEXE)

- il peut être aisément modifié afin de permettre :

- . l'accès à des objets existants dans des environnements de validité différents

- . l'intégration de nouveaux objets pédagogiques

transformation de cette relation:

- dans le sens "X est validé dans Y"

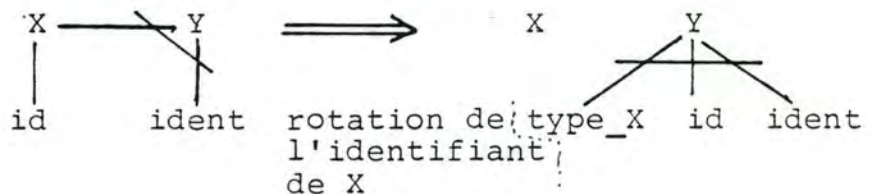
- . ce rôle de relation est constamment utilisé par l'éditeur des objets

- . l'identifiant de Y sera donc inclus dans la clé d'accès de X

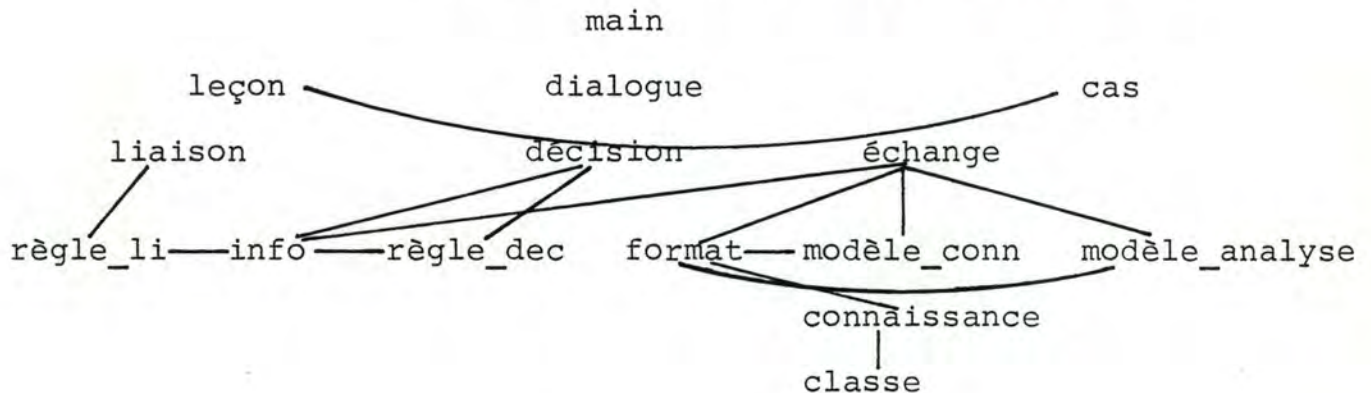
- dans le sens "Y valide X"

- . ce rôle de relation n'est pas utilisé en tant que tel

- . la détermination de l'ensemble des objets de type X valides dans Y est supportée par une structure de fenêtre d'environnement de validité, associée au type d'objet de X, réajustée à chaque changement d'environnement



. graphe de la relation "référencié-est référencié par" entre les entités principales



transformation de cette relation:

- dans le sens "X référencie Y"

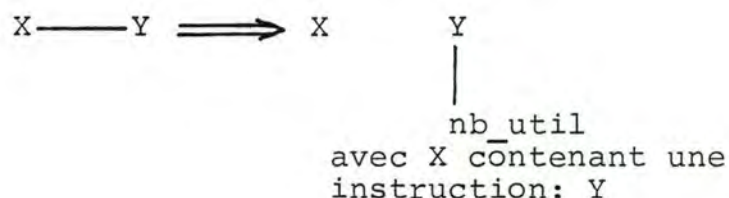
. ce rôle de relation ne se retrouve que pour indiquer que l'objet X contient une instruction ayant Y pour objet

. il n'est donc utilisé qu'à l'exécution de l'instruction

. l'identification de l'objet Y est inscrit dans le corps de l'instruction

- dans le sens "Y est référencié par X"

. la seule caractéristique retenue dans ce sens est le nombre d'utilisateurs référenciant Y: "Y est référencié par nb_util objets" ; elle permet d'autoriser une demande de suppression d'un objet ou non, selon qu'il est encore référencié ou non par d'autres objets.



21.2 Spécification des données

21.21 Définition des entités.

1. Entité : main

Il existe une seule occurrence de cette entité. Elle représente la racine des environnements des objets qui lui sont propres.

2. Entité : leçon

Une occurrence de cette entité exprime l'ensemble des enchaînements possibles de dialogues entre l'apprenant et l'ordinateur.

3. Entité : dialogue

Une occurrence de cette entité exprime l'ensemble des enchaînements possibles d'échanges élémentaires entre l'apprenant et l'ordinateur pour un dialogue déterminé.

4. Entité : liaison

Une occurrence de cette entité exprime les conditions et modalités de chaînage entre dialogues.
Les conditions portent sur l'état de l'environnement du dialogue terminé ou de la leçon courante.

5. Entité : décision

Une occurrence de cette entité exprime les conditions et modalités de chaînage entre échanges élémentaires.
Les conditions peuvent porter sur l'état global de l'environnement de l'échange terminé et du dialogue en cours.

6. Entité : échange

Une occurrence de cette entité décrit les primitives à exécuter pour réaliser un échange élémentaire de messages entre l'ordinateur et l'apprenant.

7. Entité : information

Une occurrence de cette entité décrit tout message envoyé à l'apprenant, constant ou variable, quel que soit son type ou sa localisation, dans une interaction élémentaire ou au moment d'un enchaînement entre deux interactions ou deux dialogues.

8. Entité : format

Une occurrence de cette entité décrit le mode d'acquisition d'un message en provenance de l'apprenant lors d'une interaction élémentaire : marquage, remplissage, choix multiple ou forme ouverte; dans ce dernier cas, elle décrit la forme particulière que devra prendre le message à acquérir: libre ou structuré en un ou plusieurs champs de type élémentaire.

9. Entité : modèle_analyse

Une occurrence de cette entité décrit les primitives utilisées pour effectuer l'analyse d'un message d'un format particulier.

10. Entité : modèle_connaissance

Une occurrence de cette entité exprime l'ensemble des connaissances formatées lié à une situation pédagogique déterminée.

11. Entité : connaissance

Une occurrence de cette entité décrit la connaissance associée à un format particulier d'un modèle de connaissance:

- description des classes de valeurs prévisibles, caractérisées par leur degré de pertinence, si la connaissance est locale au contexte,
- description des paramètres de la méthode globale qui engendre la connaissance si celle-ci peut être extraite d'un modèle indépendant du contexte.

12. Entité : classe

Une occurrence de cette entité décrit de façon exhaustive l'ensemble des valeurs d'un format particulier d'une classe déterminée d'une connaissance locale.

13. Entité : enviro_mess

Une occurrence de cette entité décrit les possibilités de consultation ou de requêtes, permises ou cachées à l'apprenant, lors de l'acquisition du message, pour lui permettre de réfléchir avant de répondre.

14. Entité : param_local

Une occurrence de cette entité décrit les paramètres de localisation d'un texte d'information si celle-ci est de type LOCAL sur disque.

15. Entité : descript_local

Une occurrence de cette entité décrit des caractéristiques spéciales d'un texte d'information, si celle-ci est de type LOCAL :

- si sa forme est 'remplissage', alors la caractéristique retenue est le nombre de zones libres variables du texte.
- si sa forme est 'choix multiple', la caractéristique retenue est le nombre de lignes numérotées.

16. Entité : zone_var

Une occurrence de cette entité décrit les caractéristiques d'une zone libre d'un texte.

17. Entité : info_cas

Une occurrence de cette entité décrit les paramètres permettant d'exécuter une commande d'affichage d'une partie déterminée d'un cas (fictif ou réel).

18. Entité : cas

Une occurrence de cette entité représente la description des paramètres donnant accès au dossier d'un patient (ou au dossier d'un groupe patient) fictif ou réel, d'intérêt pédagogique reconnu.

19. Entité : type_règle

Une occurrence de cette entité exprime une règle de chaînage entre deux dialogues ou deux échanges, sous la forme d'une conjonction de conditions remplies et d'un ensemble d'instructions à exécuter en cas d'évaluation positive des conditions.

20. Entité : type_conditions

Une occurrence de cette entité exprime une condition simple sous la forme d'une expression relationnelle simple du type : identifiant d'un objet $\left\{ \begin{array}{l} > \\ < \\ = \end{array} \right\} \left\{ \begin{array}{l} \text{valeur} \\ \text{identifiant} \\ \text{d'un objet.} \end{array} \right\}$

21. Entité : instruction

Une occurrence de cette entité représente la codification d'une instruction sous la forme code : identifiants numériques des paramètres associés.

22. Entité : type_champ

Une occurrence de cette entité décrit les paramètres concernant ses modalités d'affichage et le type de valeur associé d'un champ particulier d'un format structuré.

23. Entité : locale_libre

une occurrence de cette entité décrit les paramètres permettant d'analyser un message sous format libre.

24. Entité : valeurs

Une occurrence de cette entité décrit une valeur particulière d'une classe de valeurs d'un objet connaissance.

21.22 Spécification des éléments

CONVENTIONS :

- Les différents éléments à spécifier sont regroupés autour de l'entité à laquelle ils appartiennent.
- Chaque élément est décrit par les propriétés suivantes :
 - . sa description
 - . son format
 - . éventuellement sa valeur par défaut si l'élément est facultatif
 - . éventuellement ses codes si l'ensemble des codes d'un élément est limité.
 - . éventuellement son utilisation si celle-ci n'est pas standard, l'utilisation standard étant :
 valeur attribuée par l'utilisateur lors de la création du descripteur et pouvant être modifiée ultérieurement par celui-ci.
- Les éléments ont été regroupés autour des entités principales :
 - . main
 - . cas
 - . leçon
 - . dialogue
 - . liaison
 - . échange
 - . décision
 - . info
 - . format_mess
 - . modèle_analyse
 - . mod_conn
 - . connaissance
 - . valeurs

1. Entité : MAIN

Un seul type d'éléments :

- Eléments : cpt_leçon, cpt_dialogue, cpt_cas, cpt_format, cpt_info

Description : paramètres déterminant le nombre des différents sous-objets physiquement présents dans l'environnement général.

Format : numérique

Utilisation : valeurs créées par le système et mises à jour lors de toute création de nouveaux sous-objets.

2. Entité : LECON

Cinq types d'éléments :

1) Indicateur de validité

- Elément : près

Description : indicateur déterminant la cohérence logique d'une leçon.

Format : caractère.

Code : '0' : la leçon a été logiquement supprimée.

'1' : la leçon existe logiquement et est cohérente.

'2' : la leçon existe logiquement mais est incohérente.

Utilisation : valeur attribuée par le système lors de la suppression, création ou contrôle de cohérence d'une leçon.

2) Critères de sélection

- Elément : ident

Description : identifiant numérique de la leçon dans le système global.

Format : numérique

Utilisation : valeur attribuée par le système lors de la création du descripteur de la leçon
non modifiable par l'utilisateur.

- Elément : nom

Description : identifiant alphanumérique obligatoire de la leçon dans le système global.

Format : alphanumérique. [20]

- Elément : objectif

Description : qualificatif non obligatoire permettant de préciser l'objectif pédagogique poursuivi dans la leçon.

Format : alphanumérique [20]

Valeur par défaut : vide

- Elément : difficulté

Description : qualificatif non obligatoire permettant de classer les leçons poursuivant un même objectif selon leur degré de difficulté.

Format : numérique

Valeur par défaut : 0

- Elément : cas

Description : item non obligatoire identifiant le dossier d'un patient réel ou fictif servant de support à un enseignement particulier dans la leçon.

Format : numérique

Valeur par défaut : - 1

3) Dimension de l'environnement de validité des sous-objets.

- Eléments : cpt_dialogue, cpt_liaison, cpt_info

Description : paramètres déterminant le nombre des différents sous-objets physiquement présent dans l'environnement leçon.

Format : numérique

Utilisation : valeurs créées par le système et mises à jour lors de toute création de nouveaux sous-objets.

4) Indicateur d'optimisation

- Elément : optim

Description : indicateur déterminant si les accès entre les différents objets ont été optimisés ou non, une leçon étant optimisée si tous les dialogues qui la composent le sont.

Format : caractère

Code : '1' : la leçon est optimisée: tous les accès logiques ont été remplacés par des accès physiques en bytes.

'0' : la leçon n'est pas optimisée.

Valeur par défaut : '0'

Utilisation : valeur mise à '1' par le système lorsque l'utilisateur commande l'exécution du programme d'optimisation sur la leçon.

5) Amorce de la leçon

- Elément: id_liaison_intro

Description : identifiant de la liaison dans l'environnement de la leçon courante, déterminant le premier dialogue et la première liaison à activer.

Format : numérique

Valeur initiale : - 1

3. Entité : DIALOGUE

Sept types d'éléments :

1) Indicateur de validité

- Elément : pres

Description : indicateur déterminant la cohérence logique d'un dialogue.

Format : caractère

Code : '0' : le dialogue a été logiquement supprimé

'1' : le dialogue existe et est cohérent

'2' : le dialogue existe mais est incohérent.

Utilisation : valeur attribuée par le système lors de la suppression, création ou contrôle de cohérence d'un dialogue.

2) Paramètres déterminant l'environnement de validité du dialogue.

- Elément : env

Description : identifiant du type d'environnement dans lequel le dialogue courant est valide.

Format : numérique

Code : MAIN
LECON

Utilisation : valeur attribuée par le système lors de la création du dialogue, ne pouvant être modifiée ultérieurement.

- Elément : id_env

Description : identifiant de l'environnement de type env dans lequel le dialogue est valide.

Format : numérique

Utilisation : valeur attribuée par le système lors de la création du dialogue mais ne pouvant être modifiée ultérieurement.

3) Critères de sélection

- Elément : ident

Description : identifiant numérique du dialogue dans l'environnement déterminé par les paramètres env et id_env

Format : numérique

Utilisation : valeur attribuée par le système lors de la création du descripteur du dialogue, non modifiable par l'utilisateur.

- Elément : nom

Description : identifiant alphanumérique obligatoire du dialogue dans l'environnement déterminé par les paramètres env et id_env

Format : alphanumérique [20]

- Elément : objectif

Description : qualificatif non obligatoire permettant de préciser l'objectif particulier à atteindre en fin de dialogue.

Format : alphanumérique [20]

Valeur par défaut : vide

- Elément : difficulté

Description : qualificatif non obligatoire permettant de classer les dialogues poursuivant un même objectif selon leur degré de difficulté.

Format : numérique

4) Dimension de l'environnement de validité des sous-objets

- Eléments : cpt_échange, cpt_décision, cpt_info, cpt_format,
cpt_analyse_mess, cpt_mod_con, cpt_connaissance

Description : paramètres déterminant le nombre des différents sous-objets physiquement présents dans l'environnement dialogue courant.

Format : numérique

Utilisation : valeurs attribuées par le système et mises à jour lors de la création de nouveaux sous-objets.

5) Indicateur d'optimisation- Elément : optim

Description : indicateur déterminant si les accès entre les différents objets ont été optimisés ou non.

Format : caractère

Code : '1' : le dialogue est optimisé: tous les accès logiques ont été remplacés par des accès physiques en bytes.

'0' : le dialogue n'est pas optimisé.

Valeur par défaut : '0'

Utilisation : valeur mise à '1' par le système lorsque l'utilisateur commande l'exécution du programme d'optimisation sur le dialogue.

6) Indicateur du nombre d'utilisateurs de l'objet dialogue- Elément : nb_util

Description : indicateur du nombre d'utilisateurs de l'objet du dialogue.

Format : numérique

Utilisation : valeur mise à jour par le système lors de toute utilisation du dialogue par un objet liaison.

7) Amorce du dialogue- Elément : id_décision_intro

Description : identifiant de la décision dans l'environnement dialogue courant, déterminant le premier échange et la première décision à activer.

Format : numérique

Valeur initiale : - 1

4. Entité : LIAISON

Six types d'éléments :

1) Indicateur de validité

- Elément : pres

Description : indicateur déterminant la cohérence logique d'une liaison.

Format : caractère

Code : '0' : la liaison a été logiquement supprimée.

'1' : la liaison existe et est cohérente.

'2' : la liaison existe et est incohérente.

Utilisation : valeur attribuée par le système lors de la suppression, création, ou contrôle de cohérence d'une liaison.

2) Paramètres déterminant l'environnement de validité de la liaison.

- Elément : env

Description : identifiant du type d'environnement dans lequel la liaison courante est valide.

Format : numérique

Code : LECON

Utilisation : valeur attribuée par le système lors de la création de la liaison et ne pouvant être modifiée ultérieurement.

- Elément : id_env

Description : identifiant de l'environnement de type env dans lequel la liaison est valide.

Format : numérique

Utilisation : idem env

3) Critères de sélection

- Elément : ident

Description : identifiant numérique de la liaison dans l'environnement déterminé par les paramètres env et id_env

Format : numérique

Utilisation : valeur attribuée par le système lors de la création de la liaison, non modifiable par l'utilisateur.

- Elément : nom

Description : identifiant alphanumérique obligatoire de la liaison dans l'environnement déterminé par les paramètres env et id_env

Format : alphanumérique [20]

4) Dimension de l'environnement de validité des sous-objets

- Eléments : cpt_info, nb_décision

Description : paramètres déterminant le nombre des différents sous-objets valides physiquement présents dans l'environnement liaison courant.

Format : numérique

Utilisation : valeurs attribuées par le système et mises à jour lors de la création de nouveaux sous-objets (également de toute suppression de règles de décision en ce qui concerne le paramètre nb_décision).

- Elément : cpt_décision

Description : paramètre déterminant la valeur la plus élevée des identifiants de règle de décision de la liaison courante.

Format : numérique

Utilisation : valeur mise à jour par le système lors de toute création d'une nouvelle règle de décision.

5) Indicateur du nombre d'utilisateurs de la liaison

- Elément : nb_util

Description : indicateur du nombre d'utilisateurs de la liaison.

Format : numérique

Utilisation : valeur mise à jour par le système lors de toute utilisation de la liaison courante par une autre liaison.

6) Paramètres concernant les règles de décision

a) Paramètres concernant les conditions d'une règle

- Elément : nb_conditions

Description : paramètre déterminant le nombre d'expressions relationnelles simples décrivant les conditions d'application d'une règle de décision.

Format : numérique

Utilisation : valeur attribuée par le système lors de la création de l'ensemble des conditions d'une règle de décision.

, Paramètres concernant une expression simple

- Elément : env_obj

Description : identifiant du type d'environnement dans lequel se situe l'objet à comparer dans l'expression simple.

Format : numérique

Code : LECON
DIALOGUE

- Elément : obj

Description : identifiant numérique dans l'environnement de type env_obj, de l'objet à comparer dans l'expression simple.

Format : numérique

Code : (cfr. annexe : structures auxiliaires environnement_lecon, environnement_dialogue).

- Elément : rel

Description : caractère exprimant le type de relation à réaliser dans l'expression simple.

Format : caractère

Code : '<' est plus petit que
'>' est plus grand que
'=' est égal à
'c' contient (pour les structures tableaux)

- Elément : type_expr

Description : paramètre déterminant si le second membre de l'expression est une valeur ou une variable.

Format : caractère

Code : 'c' : valeur constante
'v' : variable

- Elément : valeur

Description : valeur constante ou identifiant de l'objet de référence se situant dans le même environnement que l'objet à comparer.

Format : alphanumérique [6]

b) Paramètres concernant les conséquences d'une règle

- Elément : nb_réaction

Description : paramètre déterminant le nombre d'instructions à exécuter en cas de vérification des conditions de la règle.

Format : numérique

Utilisation : valeur attribuée par le système lors de la création ou la modification de la séquence d'instruction par l'utilisateur.

• Paramètres concernant les instructions de la séquence
.....

- Elément : code

Description : identifiant du type de fonction à exécuter.

Format : caractère

Code : 'a' : afficher
't' : terminer la leçon
'r' : répéter le dialogue
'e' : exécuter le dialogue
'f' : fonctionnaliser la liaison

- Elément : type_obj

Description : identifiant numérique déterminant le type d'objet sur lequel porte l'opération de type code.

Format : numérique

Code : si code = 'a' alors INFORMATION
OBJET
EVALUATION
si code = 'e' ou 'r' alors DIALOGUE
si code = 'f' alors LIAISON

- Elément : env_obj

Description : identifiant de l'environnement de l'objet
de type type_obj sur lequel porte l'opération
de type code.

Format : numérique

Code : si type_obj = INFORMATION alors { LECON
LIAISON
si type_obj = OBJET alors { LECON
si type_obj = DIALOGUE alors { DIALOGUE
MAIN
LECON
si type_obj = LIAISON alors LECON

- Elément : obj

Description : identifiant numérique de l'objet de type
type_obj dans l'environnement env_obj, sur
lequel porte l'opération de type code.

Format : numérique

Contraintes sur les instructions d'une séquence réaction
.....

Les instructions de code 'e' et 'r' s'excluent mutuellement.
De même les instructions de code 'f' et 't'.
De plus, une instruction de code 'f' ne peut exister indépen-
damment d'une instruction de code 'e' ou de code 'r'.

5. Entité ECHANGE

Huit types d'éléments :

1) Indicateur de validité

- Elément : pres

Description : indicateur déterminant la cohérence logique d'un échange;

Format : caractère

Code : '0' : l'échange a été logiquement supprimé

'1' : l'échange existe et est cohérent

'2' : l'échange existe mais est incohérent

Utilisation : valeur attribuée par le système lors de la suppression, création ou contrôle de la cohérence d'un échange.

2) Paramètres déterminant l'environnement de validité de l'échange

- Elément : env

Description : identifiant du type d'environnement dans lequel l'échange courant est valide.

Format : numérique

Code : DIALOGUE

Utilisation : valeur attribuée par le système lors de la création de l'échange, ne pouvant être modifiée ultérieurement.

- Elément : id_env

Description : identifiant de l'environnement de type env dans lequel l'échange est valide.

Format : numérique

Utilisation : idem env

3) Critères de sélection

- Elément : ident

Description : identifiant numérique de l'échange dans l'environnement déterminé par les paramètres env et id_env

Format : numérique

Utilisation : valeur attribuée par le système lors de la création de l'échange, ne pouvant être modifiée par l'utilisateur.

- Elément : nom

Description : identifiant alphanumérique obligatoire de l'échange dans l'environnement déterminé par les paramètres env et id_env

Format : alphanumérique [20]

4) Dimension de l'environnement de validité des sous-objets

- Eléments : cpt_info, cpt_format, cpt_analyse_mess,
cpt_mod_con, cpt_connaissance

Description : paramètres déterminant le nombre des différents sous-objets valides physiquement présents dans l'environnement échange courant.

Format : numérique

Utilisation : valeurs attribuées par le système et mises à jour lors de toute création de sous-objets nouveaux.

5) Indicateur du nombre d'utilisateurs de l'échange

- Elément : nb_util

Description : nombre d'utilisateurs de l'objet échange

Format : numérique

Utilisation : valeur attribuée par le système lors de toute utilisation de l'échange par un objet décision.

6) Modèle de connaissance associé à l'échange

- Elément : env_mod_conn

Description : identifiant du type d'environnement de l'objet modèle de connaissance auquel l'échange fait référence.

Format : numérique

- Elément : id_mod_conn

Description : identifiant de l'objet modèle de connaissance dans l'environnement de type env_mod_conn

Format : numérique

7) Paramètres concernant la séquence d'instructions à exécuter- Elément : nb_inst

Description : paramètre déterminant le nombre d'instructions à exécuter.

Format : numérique

Paramètres concernant les instructions de la séquence
.....

- Elément : code

Description : identifiant du type de fonctions à exécuter lors de l'échange.

Format : caractère

Code : 'e' : envoyer l'information
 'r' : recevoir le message de l'apprenant
 'a' : analyse du message de l'apprenant.

- Elément : param

Description : identifiants numériques des types d'environnements des objets sur lesquels porte l'opération de type code ainsi que les identifiants numériques de ces objets dans ces environnements.

Si code = 'e' alors param [0]:
 . type d'environnement de l'information
 - param [1] :
 . identifiant de l'information

Si code = 'r' alors param[i] i = 0,2,4,6,8
 . types d'environnement des messages autorisés
 . les messages doivent être donnés dans l'ordre dans lequel ils doivent être analysés
 . valeur = -1 initialement.
 - param [j] j = 1,3,5,7,9
 . identifiant des messages autorisés dans les environnements de type param [i]
 i = 0,2,4,6,8

Si code = 'a' alors param[i] i = 0,2,4,6,8
 . types d'environnement des modèles d'analyse correspondant aux messages autorisés
 - param [j] j = 1,3,5,7,9
 . identifiants des modèles d'analyse dans les environnements de type param [i]
 i = 1,2,4,6,8

Format : numérique

Contrainte sur les instructions d'une séquence

Une instruction de code 'a' ne peut exister indépendamment d'une instruction de code 'r'.

8) Paramètres concernant les commandes permises à l'étudiant au cours de l'échange.

- Elément : requête

Description : identifiants des types des requêtes standard permises à l'étudiant lors de l'acquisition des messages.

Format : numérique [NOMBRE_DE_REQUETES_STANDARD]

Code : la valeur -1 indique une requête masquée.

- Elément : consult

Description : identifiants des types des consultations standard permises à l'étudiant lors de l'acquisition des messages.

Format : numérique [NOMBRE_DE_CONSULT_STANDARD]

Code : la valeur -1 indique une consultation masquée.

6. Entité DECISION

Six types d'éléments :

1) Indicateur de validité

- Elément : pres

Description : indicateur déterminant la cohérence logique de la décision.

Format : caractère

Code : '0' : la décision a été logiquement supprimée
 '1' : la décision existe et est cohérente
 '2' : la décision existe mais est incohérente.

Utilisation : valeur attribuée par le système lors de la suppression, création, ou du contrôle de cohérence d'une décision.

2) Paramètres déterminant l'environnement de validité de la décision.

- Elément : env

Description : identifiant du type d'environnement dans lequel la décision courante est valide.

Format : numérique

Code : DIALOGUE

Utilisation : valeur attribuée par le système lors de la création de la décision et ne pouvant être modifiée ultérieurement .

- Elément : id_env

Description : identifiant de l'environnement de type env dans lequel la décision est valide.

Format : numérique

Utilisation : idem env

3) Critères de sélection

- Elément : ident

Description : identifiant numérique de la décision dans l'environnement déterminé par les paramètres env et id_env

Format : numérique

Utilisation : valeur attribuée par le système lors de la création de la décision et ne pouvant être modifiée par l'utilisateur.

- Elément : nom

Description : identifiant alphanumérique obligatoire de la décision dans l'environnement déterminé par les paramètres env et id_env

Format : alphanumérique [20]

4) Dimension de l'environnement de validité des sous-objets

- Eléments : cpt_info, nb_décision

Description : paramètres déterminant le nombre des différents sous-objets valides, physiquement présents dans l'environnement décision courant.

Format : numérique

Utilisation : valeurs attribuées par le système lors de toute création de nouveaux sous-objets (et également de toute suppression de règles de décision en ce qui concerne le paramètre nb_décision).

- Elément : cpt_décision

Description : paramètre déterminant la valeur la plus élevée des identifiants de règles de décisions de l'entité décision courante.

Format : numérique

Utilisation : valeur attribuée et mise à jour par le système lors de toute création d'une nouvelle règle de décision.

5) Indicateur du nombre d'utilisateurs de la décision.

- Elément : nb_util

Description : indicateur du nombre d'utilisateurs de la décision.

Format : numérique

Utilisation : valeur attribuée et mise à jour par le système lors de toute utilisation de la décision par une autre décision.

6) Paramètres concernant les règles de décision

a) Paramètres concernant les conditions d'une règle

- Elément : nb_décision

Description : paramètre déterminant le nombre d'expressions relationnelles simples décrivant les conditions d'application d'une règle.

Format : numérique

Utilisation : valeur attribuée par le système lors de la création de l'ensemble des conditions d'une règle de décision.

. Paramètres concernant une expression simple

- Elément : env_obj

Description : identifiant du type d'environnement dans lequel se situe l'objet à comparer dans l'expression simple.

Format : numérique

Code : ECHANGE
DIALOGUE

- Elément : obj

Description : identifiant numérique dans l'environnement de type env_obj, de l'objet à comparer dans l'expression simple.

Format : numérique

Code : (cfr. annexe : structures auxiliaires environnement_échange, environnem_ent_dialogue.)

- Elément : rel

Description : caractère exprimant le type de relation à réaliser dans l'expression simple.

Format : caractère

Code : '<' est plus petit que
'>' est plus grand que
'=' est égal à
'c' contient (pour les structures tableaux)

- Elément : type_expr

Description : paramètre déterminant si le second membre de l'expression est une valeur ou une variable.

Format : caractère

Code : 'c' : valeur constante
'v' : variable

- Elément : valeur

Description : valeur constante ou identifiant de l'objet de référence se situant dans le même environnement que l'objet à comparer.

Format : alphanumérique [6]

b) Paramètres concernant les conséquences d'une règle

- Elément : nb_réactions

Description : paramètre déterminant le nombre d'instructions à exécuter en cas de vérification de la règle.

Format : numérique

Utilisation : valeur attribuée par le système lors de la création ou la modification de la séquence d'instructions par l'utilisateur.

. Paramètres concernant les instructions de la séquence
.....

- Elément : code

Description : identifiant du type de fonctions à exécuter.

Format : caractère

Code : 'a' : afficher
't' : terminer dialogue
'r' : répéter l'échange (en perdant l'environnement de l'échange terminé)
'c' : continuer avec l'échange (en conservant l'environnement de l'échange terminé)
'e' : exécuter l'échange
'p' : prendre la décision

- Elément : type_obj

Description : identifiant numérique déterminant le type d'objet sur lequel porte l'opération de type code.

Format : numérique

Code : si code = 'a' alors INFORMATION
 EVALUATION
 OBJET
 COMMENTAIRES
 REPONSE CORRECTE
 si code = 'e' ou 'r' ou 'c' alors ECHANGE
 si code = 'p' alors DECISION

- Elément : env_obj

Description : identifiant de l'environnement de l'objet
 de type type_obj sur lequel porte l'opération
 de type code.

Format : numérique

Code : si type_obj = INFORMATION alors LECON
 DIALOGUE
 DECISION
 si type_obj = OBJET alors DIALOGUE
 ECHANGE
 si type_obj = REPONSE_CORRECTE alors DIALOGUE
 ECHANGE
 si type_obj = DECISION ou ECHANGE alors DIALOGUE

- Elément : obj

Description : identifiant numérique de l'objet de type
 type_obj dans l'environnement env_obj sur
 lequel porte l'opération de type code.

Format : numérique

Contraintes sur les instructions d'une séquence réaction

Les instructions de code 'e', 'r' et 'c' s'excluent mutuellement. De même les instructions de code 'p' et 't'
 De plus, une instruction de code 'p' ne peut exister indépendamment d'une instruction de code 'e' ou 'r' ou 'c'

7. Entité INFORMATION

Six types d'éléments :

1) Indicateur de validité

- Elément : pres

Description : indicateur déterminant la cohérence logique de l'information.

Format : caractère

Code : '0' : l'information a été supprimée logiquement
 '1' : l'information existe et est cohérente
 '2' : l'information existe et est incohérente.

Utilisation : valeur attribuée par le système lors de la suppression, création, ou du contrôle de cohérence d'une information.

2) Paramètres déterminant l'environnement de validité de l'information.

- Elément : env

Description : identifiant du type d'environnement dans lequel l'information courante est valide.

Format : numérique

Code : LECON, DIALOGUE, LIAISON, ECHANGE, DECISION

Utilisation : valeur attribuée par le système lors de la création de l'environnement mais ne pouvant être modifiée ultérieurement.

- Elément : id_env

Description : identifiant numérique de l'environnement de type env dans lequel l'information est valide.

Format : numérique

Utilisation : idem env

3) Critères de sélection

- Elément : ident

Description : identifiant numérique de l'information dans l'environnement déterminé par les paramètres env et id_env

Format : numérique

Utilisation : valeur attribuée par le système lors de la création de l'information et ne pouvant être modifiée.

- Elément : nom

Description : identifiant alphanumérique obligatoire de l'information dans l'environnement déterminé par les paramètres env et id_env

Format : alphanumérique [20]

4) Indicateur du nombre d'utilisateurs de l'information

- Elément: nb_util

Description : indicateur du nombre d'utilisateurs de l'information.

Format : numérique

Utilisation : valeur mise à jour par le système lors de toute utilisation de l'information par une liaison, une décision, un échange.

5) Paramètres déterminant la présentation de l'information

- Elément : support

Description : paramètre identifiant le support physique de l'information.

Format : caractère

Code : 'd' : disque

- Elément : type

Description : paramètre identifiant le support logique de l'information.

Format : numérique

Code : si code = 'd' alors LOCAL à la leçon
MATIERE MEDICALE
REPERTOIRE
CAS

- Elément : forme

Description : paramètre déterminant la forme de présentation d'une information locale.

Format : caractère

Code : si type = LOCAL alors 'r' : texte contenant des zones variables à remplir

'c' : texte contenant des zones numérotées pour les propositions de choix multiple.

's' : texte simple

- Elément : mode_présent

Description : paramètre déterminant si une information locale utilise ou non des commandes de présentation d'écran.

Format : caractère

Code : si type = LOCAL alors 's' : texte simple, sans commandes

'c' : texte avec commandes

6) Paramètres décrivant le mode d'accès et les caractéristiques internes de l'information

a) Si_type = LOCAL

.Paramètres de localisation
.....

- Elément : adr

Description : adresse en bytes du début du texte de l'information locale, relative au début du fichier local des informations.

Format : numérique

Utilisation : valeur attribuée par le système après une validation positive de la syntaxe du texte de l'information lors de son insertion dans le fichier.

- Elément : nb_bytes

Description : nombre total de bytes dans le texte de l'information.

Format : numérique

Utilisation : idem adr

- Elément : nb_lignes

Description : nombre total de lignes dans le texte de l'information.

Format : numérique

Utilisation : idem adr

.Paramètres descriptifs
.....

- Elément : nb_l_no

Description : nombre de lignes numérotées dans le texte de l'information.

Format : numérique

Utilisation : valeur attribuée par le système après une validation positive de la syntaxe de l'information lors de son insertion dans le fichier.

- Elément : nb_zones_libres

Description : nombre total de zones libres variables dans le texte de l'information.

Format : numérique

Utilisation : idem nb_l_no

- Eléments : type, page, ligne,col, lgr

Description : paramètres décrivant le type, l'emplacement et la longueur d'une zone variable particulière.

Format : numérique

Code : pour l'élément type : ENTIER
NOMBRE REEL
CHAINE CARACTERES
SYPTOME
REMEDE
MALADIE
ANALYSE

Utilisation : idem nb_l_no

b) Si_type = CAS

- Elément : ident

Description : identifiant numérique du cas à présenter.

Format : numérique

- Elément : type_obj

Description : identifiant de la partie logique du cas à présenter.

Format : caractère

Code : 'p' : renseignements personnels
 'a' : antécédants médicaux
 'e' : examens
 'm' : maladies
 't' : tout
 'f' : renseignements personnels des membres de la
 famille.

- Elément : commande

Description : zone permettant de décrire la commande d'accès
 à une partie de l'objet de type type_obj

Format : alphanumérique [80]

8. Entité FORMAT_MESSAGE

Six types d'éléments :

1) Indicateur de validité

- Elément : pres

Description : indicateur déterminant la cohérence logique du format.

Format : caractère

Code : '0' : le format a été supprimé logiquement

'1' : le format existe et est cohérent

'2' : le format existe mais est incohérent

Utilisation : valeur attribuée par le système lors de la suppression, création ou du contrôle de cohérence d'un format.

2) Paramètres déterminant l'environnement de validité du format

- Elément : env

Description : identifiant du type d'environnement dans lequel le format est valide.

Format : numérique

Code : DIALOGUE
ECHANGE

Utilisation : valeur attribuée par le système lors de la création du format mais ne pouvant être modifiée ultérieurement.

- Elément; id_env

Description : identifiant numérique de l'environnement de type env dans lequel le format est valide.

Format : numérique

Utilisation : idem env

3) Critères de sélection

- Elément : ident

Description : identifiant numérique du format dans l'environnement déterminé par les paramètres env et id_env

Format : numérique

Utilisation : valeur attribuée par le système lors de la création du format, et ne pouvant être modifiée.

- Elément : nom

Description : identifiant alphanumérique obligatoire du format dans l'environnement déterminé par les paramètres env et id_env

Format : alphanumérique [20]

4) Indicateur du nombre d'utilisateurs du format

- Elément :nb_util

Description : indicateur du nombre d'utilisateurs du format.

Format : numérique

Utilisation : valeur mise à jour par le système lors de toute utilisation du format par un échange.

5) Paramètres descriptifs du format

- Elément : mode

Description : paramètre déterminant le mode d'acquisition du message.

Format : caractère

Code : 'm' : marquage
'r' : remplissage
'c' : choix multiple
'o' : mode de la forme ouverte

- Elément:forme

Description : paramètre déterminant la forme que doit prendre le message en cas d'acquisition de messages ouverts.

Format : caractère

Code : si mode = 'o' alors 'l' : forme libre
's' : forme structurée en un ou plusieurs champs.

- Elément : nb_lignes

Description : nombre de lignes que peut comporter une valeur associée au format.

Format : numérique

Code : valeur maximum : 2 si le format ne contient aucun champ de type symptôme

:11 si le format contient un champ de type symptôme.

6) Paramètres concernant les champs du format

- Elément ; nb_champs

Description : paramètre déterminant le nombre de champs dans un format.

Format : numérique

. Paramètres descriptifs des différents champs

- Elément : nom

Description : mnémonique caractérisant un champ particulier d'un format structuré.

Format : alphanumérique [10]

Valeur par défaut : vide

- Eléments : ad_rel, type, lgr, ligne, col, importance

Description : paramètres déterminant :

- l'adresse relative de la valeur du champ par rapport au début du format
- le type élémentaire de valeur
- la ligne, la colonne et la longueur maximum que les valeurs de ce champ devront occuper à l'écran
- l'importance du champ à l'intérieur du format (afin de déterminer l'ordre de leur évaluation).

Format : numérique

Remarque : l'élément ligne n'est utilisé que si le format s'étend sur deux lignes: il indique alors le numéro de la ligne relative.

9. Entité MODELE_CONNAISSANCE

Six types d'éléments :

1) Indicateur de validité

- Elément : pres

Description : indicateur déterminant la cohérence logique du modèle de connaissance.

Format : caractère

Code : '0' : le modèle de connaissance a été supprimé logiquement.

'1' : le modèle de connaissance existe et est cohérent

'2' : le modèle de connaissance existe et est incohérent.

Utilisation : valeur attribuée par le système lors de la suppression, création, ou du contrôle de cohérence d'un modèle de connaissance.

2) Paramètre déterminant l'environnement de validité du modèle de connaissance

- Elément: env

Description : identifiant du type d'environnement dans lequel le modèle de connaissance est valide.

Format : numérique

Code : ECHANGE
DIALOGUE

Utilisation : valeur attribuée par le système lors de la création du modèle de connaissance mais ne pouvant être modifié ultérieurement.

- Elément : id_env

Description : identifiant numérique de l'environnement de type env dans lequel le modèle de connaissance est valide.

Format : numérique

Utilisation : idem env

3) Critères de sélection

- Elément : ident

Description : identifiant numérique du modèle de connaissance dans l'environnement déterminé par les paramètres env et id_env

Format : numérique

Utilisation : valeur attribuée par le système lors de la création du modèle de connaissance et non modifiable ultérieurement.

- Elément : nom

Description : identifiant alphanumérique obligatoire du modèle de connaissance dans l'environnement déterminé par les paramètres env et id_env

Format : alphanumérique [20]

4) Indicateur du nombre d'utilisateurs du modèle de connaissance

- Elément: nb_util

Description : indicateur du nombre d'utilisateurs du modèle de connaissance.

Format : numérique

Utilisation : valeur mise à jour par le système lors de toute utilisation du modèle de connaissance par un échange.

5) Structure du modèle de connaissance

- Elément : structure

Description : paramètre déterminant l'organisation des différentes connaissances associées à une même situation pédagogique, chaque connaissance étant caractérisée par un format particulier.

Format : numérique

Code : ENSEMBLE

- Elément : nb_format_util

Description : paramètres déterminant le nombre de formats différents utilisés, chacun d'eux caractérisant une connaissance particulière.

Format : numérique

- Elément env_format_util

Description : identifiants des environnements des formats utilisés.

Format : numérique [5]

- Elément format_util

Description : identifiants numériques des formats utilisés dans les environnements correspondants déterminés par env_format_util.

Format : numérique [5]

10. Entité CONNAISSANCE

Sept types d'éléments

1) Indicateur de validité

- Elément: pres

Description : indicateur déterminant la cohérence logique de la connaissance

Format : caractère

Code : '0' : la connaissance a été supprimée logiquement
 '1' : la connaissance existe et est cohérente
 '2' : la connaissance existe mais est incohérente

Utilisation : valeur attribuée par le système lors de la suppression, création, ou contrôle de cohérence d'une connaissance.

2) Paramètre déterminant l'environnement de validité de la connaissance

- Elément : env

Description : identifiant du type d'environnement dans lequel la connaissance est valide.

Format : numérique

Code : MODELE_CONN

Utilisation : valeur attribuée par le système lors de la création de la connaissance.

- Elément id_env

Description : identifiant numérique de l'environnement de type env dans lequel la connaissance est valide.

Format : numérique

Utilisation : idem env

3) Critères de sélection

- Elément : ident

Description : identifiant numérique de la connaissance dans l'environnement déterminé par les paramètres env et id_env

Format : numérique

Utilisation : valeur attribuée par le système lors de la création de la connaissance, non modifiable ultérieurement.

- Elément : nom

Description : identifiant alphanumérique obligatoire de la connaissance dans l'environnement déterminé par les paramètres env et id_env

Format : alphanumérique [20]

4) Identifiant interne

- Elément : id_int

Description : identifiant interne de la connaissance basée sur les paramètres env, id_env et ident

Format : numérique

Utilisation : idem env

5) Indicateur du nombre d'utilisateurs de la connaissance

- Elément : nb_util

Description : indicateur du nombre d'utilisateurs de la connaissance.

Format : numérique

Utilisation : valeur attribuée par le système (actuellement non modifiée).

6) Identifiant du format associé, du ou des formats utilisés

- Eléments : env_format

Description : identifiant(s) du (des) environnement(s) du (des) format(s) utilisé(s) dans la connaissance.

env_format [0] : concerne le format de la connaissance

env_format [i] i = 1, 2 : concernent les formats utilisés comme paramètres du modèle si la connaissance est globale.

Format : numérique [3]

- Eléments : id_format

Description : identifiants dans les environnements correspondant env_format, des formats utilisés.

Format : numérique [3]

7) Paramètres décrivant la connaissance

- Elément : forme

Description : paramètre déterminant si la connaissance est locale ou globale.

Format : caractère

Code : 'l' : locale
'g' : globale

+ Si_forme = 'l' et forme du format associé = 'l' :

- Elément : commande

Description : ligne de commande utilisée dans le cas où un ordre particulier et des nuances de pertinence des mots-clés sont demandés dans l'analyse d'un message libre.

Format : alphanumérique [80]

- Elément : nb_cl_mots

Description : paramètre déterminant le nombre de classes de mots-clés de référence.

Format : numérique

- Elément : conversion

Description : identifiant d'une fonction standard de conversion du message avant l'analyse.

Format : numérique

Code : 'b' : suppression des séparateurs superflus

+ Si_forme = 'l' et forme du format associé = 'e' ou 's' :

. Paramètres concernant l'ensemble des classes
.....

- Elément nb_classe

Description : paramètre déterminant le nombre de classes de valeurs de référence.

Format : numérique

- Elément cpt_classe

Description : paramètre déterminant la plus haute valeur des identifiants d'une classe de valeurs dans la connaissance courante.

Format : numérique

Utilisation : valeur mise à jour par le système lors de la création d'une nouvelle classe.

- Elément : nb_elem

Description : paramètre déterminant le nombre total des valeurs de toutes les classes de la connaissance.

Format : numérique

- Elément : id_classe_correcte

Description : identifiant de la classe des réponses correctes de la connaissance.

Format : numérique

.Paramètres descriptifs des classes de valeurs
.....

- Elément: nom

Description : mnémonique non obligatoire caractérisant la classe de valeurs.

Format : alphanumérique [10]

- Elément : ident

Description : identifiant numérique de la classe dans l'environnement connaissance.

Format : numérique

Utilisation : valeur attribuée par le système lors de la création de la classe; non modifiable.

- Elément : nuance

Description : degré de pertinence des valeurs de la classe dans l'analyse du message.

Format : caractère

Code : 'c' : correct et obligatoire
't' : tolérable non obligatoire
'i' : incorrect

- Elément: id_commen

Description : identifiant numérique dans l'environnement dialogue, du commentaire à afficher en cas de proposition par l'apprenant d'au moins une des valeurs de la classe.

Format : numérique

- Elément nb_elem

Description : nombre total de valeurs dans la classe courante.

Format : numérique

+ Si_forme = 'g' _

- Elément : code

Description : identifiant du modèle ou de la primitive à exécuter pour engendrer la connaissance.

Format : numérique

Code : . si le format déterminé par env_format [0]
et id_format [0]

est de forme élémentaire et de type remède
alors FREQUENCE
DEGRE
ELECTRE

. si le format déterminé par env_format [0]
et id_format [0]

est de forme élémentaire et de type symptôme
alors DIAGNOSTIC_DIFF
PATHOGENESIE
MOTS_CLES

- Elément : param

Description : -param [0] : identifiant de l'environnement du format des valeurs des paramètres d'entrée.

-param [1] : identifiant du format, dans l'environnement param [0], des valeurs des paramètres d'entrée.

-Remarque: le format identifié par param [0] et param [1] doit se retrouver dans le modèle de connaissance dont la connaissance fait partie.

-Si code = ELECTRE :

param [3]: choix parmi les méthodes standard concernant les seuils de concordance.

param [4]: choix parmi les méthodes standard concernant les seuils de discordance.

param [5]: choix parmi les méthodes standard concernant les poids des symptômes

Format : numérique

11. Entité : MODELE_ANALYSE

Six types d'éléments :

1) Indicateur de validité

- Elément : pres

Description : indicateur déterminant la cohérence logique du modèle d'analyse.

Format : caractère

Code : '0' : le modèle d'analyse a été logiquement supprimé
 '1' : le modèle d'analyse existe et est cohérent
 '2' : le modèle d'analyse existe et est incohérent

Utilisation : valeur attribuée par le système lors de la suppression, création, ou contrôle de cohérence du modèle d'analyse.

2) Paramètres déterminant l'environnement de validité du modèle d'analyse

- Elément : env

Description : identifiant du type d'environnement dans lequel le modèle d'analyse est valide.

Format : numérique

Code : ECHANGE
 DIALOGUE

Utilisation : valeur attribuée par le système lors de la création du modèle d'analyse et non modifiable ultérieurement.

- Elément id_env

Description : identifiant numérique de l'environnement de type env dans lequel le modèle d'analyse est valide.

Format : numérique

Utilisation : idem env

3) Critères de sélection

- Elément : ident

Description : identifiant numérique du modèle d'analyse dans l'environnement déterminé par les paramètres env et id_env

Format : numérique

Utilisation : valeur attribuée par le système lors de la création du modèle d'analyse, et non modifiable.

- Elément : nom

Description : identifiant alphanumérique obligatoire du modèle d'analyse dans l'environnement déterminé par les paramètres env et id_env

Format : alphanumérique [20]

4) Indicateur du nombre d'utilisateurs du modèle d'analyse

- Elément : nb_util

Description : indicateur du nombre d'utilisateurs du modèle d'analyse.

Format : numérique

Utilisation : valeur mise à jour par le système lors de toute utilisation du modèle d'analyse par un échange.

5) Formats utilisés

- Elément : nb_format_util

Description : nombre de formats auxquels on fait référence dans les primitives du modèle d'analyse.

Format : numérique

- Elément : env_format

Description : identifiant de l'environnement du format utilisé.

Format : numérique [3]

- Elément : id_format

Description : identifiant du format dans l'environnement correspondant env_format.

Format : numérique [3]

6) Description de s primitives

- Elément : code

Description : identifiant du type de fonction à exécuter.

Format : caractère

Code : 'c' : comparaison
 'm' : modèle
 'e' : évaluation

- Elément : fonction

Description : identification numérique de la fonction de type code à exécuter.

Format : numérique

Code : si code = 'm'

alors (actuellement)

. si le format déterminé par env_format [0]
et id_format [0]

est de forme élémentaire et de type remède

alors FREQUENCE

DEGRE

ELECTRE

. si le format déterminé par env_format [0]
et id_format [0]

est de forme élémentaire et de type symptôme

alors DIAGNOSTIC_DIFF

PATHOGENESIE

MOTS_CLES

si code = 'c' alors COMP_CLASSE
COMP_DIFF

COMP_LIBRE

si code = 'e' alors EVAL_STD
EVAL_CERT

- Eléments: paramètres

Description : si code = 'm' alors

param [0] = identifiant de l'environnement du format des valeurs des paramètres d'entrée.

param [1] = identifiant du format dans l'environnement param [0], des valeurs des paramètres d'entrée.

param [2] = classe de réponses à utiliser comme paramètres d'entrée.
code : CORRECTES
APPRENANT

si fonction = ELECTRE alors

param [3] = choix parmi les méthodes standards concernant les seuils de concordance

param [4] = choix parmi les méthodes standards concernant les seuils de discordance.

param [5] = choix parmi les méthodes standards concernant les poids des symptômes.

. si code = 'c' et fonction = COMP_DIFF, alors
 param [0] et param [5] =
 identifiants des environnements des formats
 des valeurs à comparer et de référence.
 param [1] et param [6] =
 identifiants des formats dans les environ-
 nements param 8 et param [5], des valeurs
 à comparer et de référence.
 param [3] et param [8] =
 classe de réponses à utiliser comme valeurs
 à comparer et de référence
 Codes: param [2] : APPRENANT
 param [7] : CORRECTES
 GENEREES
 param [2] et param [7] =
 adresse relative dans leur format, des
 valeurs à comparer et de référence.
 param [4] et param [9] =
 dimensions des classes de valeurs à comparer
 et de référence.

Format : numérique

12. Entité : VALEURS

Trois types d'éléments :

1) Indicateur de validité :

- Elément : pres

Description : indicateur déterminant la cohérence logique de la valeur.

Format : numérique

Code : '0' : la valeur a été supprimée logiquement
'1' : la valeur existe.

Utilisation : valeur attribuée par le système lors de la suppression ou création d'une classe de valeurs.

2) Paramètres d'identification

- Elément : id_conn

Description : identifiant numérique interne de la connaissance qui fait référence à cette valeur.

Format : numérique

- Elément id_classe

Description : identifiant numérique de la classe dans l'environnement id_conn, contenant cette valeur.

Format : numérique

- Elément : id

Description : identifiant numérique d'un groupe de valeurs synonymes.

Format : numérique

- Elément : syn

Description : identifiant numérique de la valeur dans le groupe de valeurs synonymes.

Format : numérique

3) Valeur

- Elément : valeur

Description : zone contenant la valeur

Format : alphanumérique [120]

13. Entité : CAS

Trois types d'éléments :

1) Indicateur de validité

- Elément : pres

Description : indicateur déterminant la cohérence logique du cas.

Format : numérique

Code : '0' : le cas a été logiquement supprimé
'1' : le cas existe.

Utilisation : valeur attribuée par le système lors de la suppression ou création d'un cas.

2) Critères de sélection

- Elément : ident

Description : identifiant numérique du cas dans l'environnement global.

Format : numérique

Utilisation : valeur attribuée par le système lors de la création du cas.

- Elément : nom

Description : identifiant alphanumérique du cas dans l'environnement global.

Format : alphanumérique [20]

- Elément : objectif

Description : le ou les mots-clés décrivant le cas envisagé dans le dossier.

Format : alphanumérique [20]

- Elément : type

Description : paramètre déterminant si un cas est réel ou fictif.

Format : caractère

Code : 'r' : réel
'f' : fictif

3) Description du contenu du cas

- Elément : no_doss

Description : identifiant du numéro de dossier si le cas
est issu de la base des dossiers centralisés.

Format : numérique

- Elément : id_doc

Description : identifiant numérique de l'information glo-
bale décrivant brièvement l'intérêt du cas.

Format : numérique

II.22 - SPECIFICATION DES TRAITEMENTS

22.1 METHODE

22.11 Identification des traitements

L'ensemble des traitements du système d'information à automatiser peut être décomposé en plusieurs étapes selon des critères de hiérarchisation différents :

1. Première étape : décomposition conceptuelle (cfr [5] p. 50)

Repères retenus dans la décomposition :

- Projet : concerne la partie du système d'information à automatiser délimité par les objectifs opérationnels et stratégiques associés à ce projet.
- Applications : concernent des traitements quasi autonomes du projet.
- Phases : le critère d'identification retenu est celui de l'unité d'exécution et de l'unité des ressources du système d'information.
- Fonctions : traitement élémentaire d'une sémantique simple.

2. Deuxième étape : décomposition organique

La décomposition organique des traitements retenus dans la première étape met en évidence de nouveaux modules (primitives fonctionnelles de gestion fichier et de gestion écran). L'ensemble de ces modules peut être hiérarchisé de façon à cacher le plus possible d'informations procédurales à l'intérieur de chaque module.

22.12 Relations entre traitements

- La relation qui relie plusieurs traitements dans la décomposition conceptuelle est de type 'INVOQUE'; un traitement en invoque un autre mais tous deux garde le même contexte global d'exécution.
- Les traitements issus de la décomposition conceptuelle sont considérés comme modules de haut niveau dans la décomposition organique. Ils sont reliés aux nouveaux modules par la relation 'UTILISE' : un module utilise un module de niveau inférieur de la hiérarchisation, les informations du contexte d'exécution de chacun d'eux étant quasi autonomes.

22.13 : Démarche adoptée

Pour la spécification des traitements, les deux étapes ont été unifiées :

- Les traitements issus de la décomposition fonctionnelle peuvent être considérés comme faisant partie d'un niveau '0' dans la hiérarchie selon la relation 'UTILISE'. Ces traitements seront spécifiés par les caractéristiques suivantes :

objectif
arguments
résultats
invoque
est invoqué par
procédure

- Les traitements issus de la décomposition organique, autres que les traitements de niveau conceptuel, seront situés dans leur niveau de hiérarchisation. Leurs caractéristiques procédurales n'ont pas été spécifiées dans le cadre de cette démarche. Ils sont caractérisés par la liste de leurs arguments dont la nomenclature en détermine la fonction.

22.2 Niveau 'UTILISE' 0 -----

22.21 Projet

Objectif :

- développer un système d'aide à l'enseignement dans le cadre de la formation des médecins homéopathes, plus particulièrement dans les domaines d'initiation à la technique répertoriale et de la pratique du diagnostic et de la thérapeutique homéopathique.
- concevoir, gérer, valider la syntaxe des objets composant une leçon.
- exécuter une leçon en mode conversationnel.
- gérer les traces des exécutions pour effectuer la validation sémantique des leçons créées.

Reçoit :

- demande de conception, d'exécution ou d'analyse des traces d'exécution.

Génère :

objet_édité
objet_valide
objet_incohérent

Invoke :

édition_leçon
exécution_leçon
suivi_leçon
optimisation_leçon.

22.22 Applications

Application 1 : EDITEUR_LECON

Objectif :

Son but est de fournir à l'utilisateur-concepteur de la leçon :

- des commandes lui permettant de décrire les différents dialogues et les liaisons entre ces dialogues afin d'atteindre un objectif pédagogique particulier, mais aussi de détruire, modifier, afficher, tester, éditer, les différents objets qu'il a déjà créés, d'interrompre une opération en cours pour consulter d'autres objets.
- une aide à la conception des leçons :
 - en proposant :
 - . des menus au niveau de chaque étape de chaque commande;
 - . sur demande, des messages explicitant le choix des possibilités offertes à chaque niveau;
 - en affichant en permanence l'état de l'environnement dans lequel le concepteur travaille;
 - en testant à chaque niveau la cohérence syntaxique des différents objets créés.

Reçoit :

demande_conception_leçon

Génère :

rien OU
 leçon_éditée OU/ET
 dialogue_édité OU/ET
 info_éditée OU/ET
 leçon_cohérente OU/ET
 dialogue_cohérent OU/ET
 leçon_incohérente OU/ET
 dialogue_incohérent OU/ET

Invoque :

init_environt
 gestion_commande
 clôture_environt.

Application 2 : EXECUTION_LECON

Objectif :

Son but est :

- de supporter les échanges entre le logiciel de formation du médecin et l'apprenant;
- de présenter des connaissances particulières à l'apprenant;
- de procurer à l'apprenant des outils, de la documentation à consulter pour l'aider à répondre;
- d'évaluer les réponses de l'apprenant selon des modèles d'analyse de connaissance déterminés;
- d'acquérir les réponses de l'apprenant sous forme libre ou structurée, élémentaire ou groupée, par remplissages de zones vierges dans un texte, par soulignements des éléments principaux d'un texte, par sélection dans un choix multiple de réponses;
- de donner des éléments plus ou moins détaillés de l'évaluation des réponses et de diriger l'enchaînement des échanges ou des dialogues en fonction des résultats des évaluations, des outils utilisés, du nombre des essais effectués, du résultat global, ...
- de conserver une trace de l'ensemble des dialogues et des résultats des évaluations.

Reçoit :

demande_exécution_leçon

Génère :

trace_leçon
trace_dialogue
trace_échange
trace_valeurs

Invoque :

init_environnements
gestion_dialogue
cloture_environnements

Application 3 : SUIVI DES LECONS

Objectif :

Son but est de fournir à l'écran ou à l'imprimante, des états statistiques concernant les différentes exécutions des leçons et dialogues sur la base des traces relevées lors des expérimentations de ces leçons auprès d'étudiants.

Reçoit :

demande_de_statistiques

Génère :

rien OU
édition_statistiques

Invoke :

initialisations
gestion_cde_stat
clôtures

Application 4 : OPTIMISATION_LECON

Objectif :

Lorsqu'une leçon et les dialogues qui la constituent sont définitivement acceptés, des optimisations au niveau des accès aux différents objets peuvent être réalisés pour améliorer le temps de réponse à l'exécution.

Ces objets seront dépouillés des informations devenues inutiles (compteurs, identifiants des environnements, nombre d'utilisateurs, ...)

Si certains des objets utilisés dans les leçons ou dialogues font partie de l'environnement global, une copie de ces objets restera dans la base des objets non optimisés, sinon ils y seront détruits.

Reçoit :

demande_optim_leçon
demande_optim_dialogue

Génère :

leçon_optimisée
dialogue_optimisé

Invoque :

init_environt
gestion_optimisation
clôture_environt

Remarque : cette application est une extension possible du système et n'a pas été spécifiée.

22.23 Phases

CONVENTION

- les phases ont été regroupées par application.
- pour chaque application :
 - 1) les modules coordinateurs de phases
 - 2) les phases groupées par objet
 - 3) les phases communes à plusieurs objets.

1. APPLICATION 1

11. Modules de coordination

- 111. init_environt
- 112. gestion_commande
- 113. clôture_environt

111 : init_environt

Objectif : initialiser le contexte d'exécution de l'éditeur d'objets pédagogiques.

Reçoit : demande_conception_leçon

Génère : contexte_initialisé

Procédure: init_écran
définition et ouverture fichier
init_environnement.

112 : gestion_commande

Objectif : acquérir, valider et exécuter une commande.

Reçoit : contexte_initialisé OU
fin_commande

Génère : fin_commande OU
fin_travail

Invoke : get_command_op
get_command_obj
get_command_sélection
oper [opération] [objet] (cfr tableau)

TABLEAU DES OPERATIONS POSSIBLES SUR LES DIFFERENTS OBJETS
===== oper [opération] [objet] =====

	CREATION	MODIFICATION	SUPPRESSION	AFFICHAGE	LISTAGE	EDITION	ENREGISTREMENT	SELECTION	INTERRUPTION
1. CAS	création_ cas	modification_ cas	suppression_ cas	affichage_ cas	lister	édition_ cas	enregistrement_ cas	sélection	0
2. LECON	création_ leçon	modification	suppression	affichage	lister	édition	enregistrement	sélection	interruption
3. DIALOGUE	création_ dialogue	modification	suppression	affichage	lister	édition	enregistrement	sélection	interruption
4. LIAISON	création_ liaison	modification	suppression	affichage	lister	0	enregistrement	sélection	0
5. ECHANGE	création_ échange	modification	suppression	affichage	lister	0	enregistrement_ échange	sélection	interruption
6. DECISION	création_ décision	modification	suppression	affichage	lister	0	enregistrement	sélection	0
7. INFORMATION	création_ info	modification_ info	suppression	affichage_ info	lister	édition	enregistrement	sélection	0
8. FORMAT	création_ format	0	suppression	affichage_ format	lister	0	enregistrement	sélection	0
9. MODELE CON_ NAISSANCE	création_ mod_con	modification	suppression	affichage	lister	0	enregistrement	sélection	0
10. CONNAISSANCE	création_ connaiss	modification_ connaissance	suppression_ connaissance	affichage_ connaiss	lister	0	enregistrement	sélection	0
11. CLASSE_VALEURS	création_ classe	modification_ classe	suppression_ classe	affichage_ classe	lister_ classes	0	enregistrement_ classe	sélection	0
12. MODELE_ANALYSE	création_ mod_anal.	modification_ mod_analyse	suppression	affichage_ mod_anal.	lister	0	enregistrement_ mod_analyse	sélection	0
13. REGLE_LIEN_DEC	création_ lien_dec	modification_ lien_dec	suppression_ lien_dec	affichage_ lien_dec	0	0	enregistrement_ lien_dec	sélection	0
14. REGLE_LIEN_ LIAISON	création_ lien_liais	modification_ lien_liaison	suppression_ lien_liaison	affichage_ lien_liais	0	0	enregistrement_ lien_liaison	sélection	0

Procédure :

- . initialiser les fenêtres des environnements de validité des sous-objets de l'environnement courant.
- . acquérir_valider_opération
- . Si opération \neq FIN et opération \neq ANNULATION
alors :
 acquérir_valider_objet
 Si objet \neq ANNULATION
 alors :
 acquérir_valider_sélection

 Si identifiant_objet \neq ANNULATION
 alors :
 Si opération = MODIFICATION
 alors : acquérir_valider_sélection
 Si identifiant_objet \neq ANNULATION
 alors : valider_oper [opération] [objet]

 Si opération valide
 alors : .exécuter oper [opération] [objet]
 .mettre à jour l'environnement
 sinon : revenir à l'environnement père

 Fin_commande.
- . Si opération = FIN
 alors : fin_travail
 sinon : annuler_commande
 fin_commande

113 : clôture_environt

- objectif : clôturer le travail de conception
- Reçoit : fin_travail
- Génère : fin_conception_leçon
- Procédure : compacter les fichiers (éliminer les descripteurs supprimés logiquement)
 fermer les fichiers.

12. Phases proprement dites groupées par objet

12.1 : Objet cas

- Phase création_cas

Objectif : préparer la création d'un cas de dossier
réel ou fictif

Reçoit : descripteur de cas

Génère : cas_créé
cas_annulé

Invoke : ss_proc

Procédure: si type_cas = 'd'
alors : créer fichier_commande avec commande
= DS
sinon : créer fichier_commande avec commande
= CC

- .exécuter en sous-processus le programme
gestion_cas
- extraire du fichier renvoyé le numéro de
dossier attribué
 - . si numéro de dossier est négatif
alors annuler la commande
 - . sinon enregistrer_cas.

- Phase modification_cas

Objectif : préparer la modification d'un cas de dossier
réel ou fictif.

Reçoit : descripteur de cas

Génère : cas_modifié OU
cas_annulé

Invoke : ss_proc

Procédure: si type_cas = 'd'
alors : message_erreur
sinon :
- créer fichier_commande avec commande
= CM no_dos
- exécuter en sous-processus le programme
gestion_cas
- extraire du fichier résultat un indicateur
- si indicateur est négatif,
alors : cas annulé
sinon : enregistrer_cas
cas_modifié.

- Phase suppression_cas

Objectif : préparer la suppression d'un cas de dossier fictif ou réel.

Reçoit : descripteur_cas

Génère : cas_supprimé OU
cas_valide.

Invoque : ss_proc

Procédure: - si nb_util > 0
 alors : nb_util = nb_util - 1
 message_erreur
 sinon :
 - si type_cas = 'd'
 alors : message_erreur
 sinon : - créer fichier_commande avec com-
 mande = CS no_dos
 - exécuter en sous-processus le
 programme gestion_cas
 - extraire du fichier renvoyé un
 indicateur
 - si indicateur est négatif
 alors : cas annulé
 sinon : cas_supprimé

- Phase affichage_cas

Objectif : préparer l'affichage d'un cas de dossier réel ou fictif.

Reçoit : descripteur de cas

Génère : cas_affiché

Invoque : ss_proc

Procédure: - si type_cas = 'd'
 alors : créer fichier_commande avec commande
 = DA no_dos
 sinon : créer fichier_commande avec commande
 = CA no_dos
 - exécuter en sous-processus le programme de
 gestion_cas
 - cas affiché

123 : Objet dialogue- Phase création_dialogue
-----Objectif : préparer la création d'un dialogueReçoit : descripteur de dialogueGénère : dialogue_crééInvoque : /Procédure : initialiser les attributs :

optim = '0'

id_décision_intro = -1

cpt_échange = cpt_décision = cpt_info = cpt_format

= cpt_analyse_mess = cpt_mod_conn =

cpt_connaissance = nb_util = 0

124 : Objet liaison- Phase création_liaison
-----Objectif : préparer la création d'un ensemble de règles de chaînage entre les dialogues.Reçoit : descripteur de liaisonGénère : /Invoque : /Procédure : initialisation des attributs :

cpt_info = cpt_décisions = nb_décisions =

nb_réactions = nb_util = 0

- Phase suppression_liaison
-----Objectif : supprimer une liaison si elle n'est pas utilisée par une autre liaison.Reçoit : descripteur de liaisonGénère : liaison_suppriméeInvoque : suppressionProcédure : si nb_util > 0

alors nb_util = nb_util - 1

message_erreur

sinon : suppression

- Phase création_intro_leçon

Objectif : créer la première liaison d'une leçon.

Reçoit : /

Génère : liaison_créée

Invoke : création_liaison
défaut_sélect

Procédure : - paramètres de sélection par défaut pour la
liaison
- création_liaison
- id_intro_leçon de la leçon courante =
ident de la liaison créée.
- positionner indicateur intro_liaison.

- Phase suppression_intro_leçon

Objectif : supprimer la première liaison d'une leçon

Reçoit : /

Génère : liaison_supprimée

Invoke : suppression_liaison

Procédure : - accéder à la liaison id_intro_leçon
- id_intro_leçon de la leçon courante = -1
- suppression_liaison.

125 : Objet échange

- Phase création_échange

Objectif : préparer la création des sous-objets d'un échange.

Reçoit : descripteur d'échange

Génère : échange_créé

Procédure : initialiser les attributs :
nb_util = nb_instr = 0
compteur = 0

- Phase création_séquence_échange

Objectif : créer la séquence des instructions à exécuter lors d'une interaction élémentaire.

Reçoit : descripteur d'échange

Génère : séquence_échange_créé
séquence_échange_annulé

Procédure : - acquisition_objet modèle de connaissance référencié

- tant que pas fin_séquence ou annulation :
acquérir commande d'édition
si cde = insérer :
alors : . créer_instruction_échange
 . incrémenter le compteur d'utilisation
 de l'objet de l'opération
 . incrémenter nb_instr

si cde = détruire
alors : . détruire_instruction_échange
 . décrémenter le compteur d'utilisation
 de l'objet de l'opération
 . décrémenter nb_instr

- acquérir suite des requêtes autorisées

- acquérir suite des consultations autorisées.

Règle de construction d'une instruction :
cfr. Deuxième partie II-21.22:Spécification des
éléments. p 74 s.s.

- Phase modification_séquence_échange

Objectif : ajouter ou supprimer une instruction dans la séquence d'un échange.

Reçoit : descripteur d'échange

Génère : échange_modifié
échange_annulé

Procédure : elle est identique à celle de la phase création_échange, les initialisations étant omises.

- Phase affichage_séquence_échange

Objectif : afficher l'énoncé de la suite des instructions.

Reçoit : descripteur de l'échange

Génère : séquence_échange_afficher

Procédure : pour chaque instruction
 . décodé code opération
 . accéder aux objets concernés
 . afficher prompt_objet
 . afficher nom des objets.

- Phase suppression_séquence_échange

Objectif : supprimer la séquence d'instructions attachée
 à l'échange.

Reçoit : descripteur de l'échange

Génère : séquence_échange_supprimée

Procédure : nb_instr = 0

126 : Objet décision

- Phase création_décision

Objectif : préparer la création des règles de chaînage
 entre les échanges.

Reçoit : descripteur de décision

Génère : décision créée

Procédure : initialiser attributs :
 cpt_info = nb_décision = cpt_décision = nb_util = 0

- Phase suppression_décision

Objectif : supprimer une liaison si elle n'est pas utilisée
 par une autre décision.

Reçoit : descripteur de décision

Génère : décision_supprimée

Invoke : suppression

Procédure : si nb_util > 0
 alors : nb_util = nb_util -1
 message_erreur
 sinon : suppression

- Phase création_intro_dialogue

Objectif : créer la première décision du dialogue
Reçoit : /
Génère : décision créée
Invoque : création_décision
Procédure : - paramètres de sélection par défaut pour la
 décision
 - création_décision
 - id_intro_dialogue du dialogue = ident de la
 décision créée,
 - positionner indicateur intro_dial

- Phase suppression_intro_dialogue

Objectif : supprimer la première décision d'un dialogue
Reçoit : /
Génère : décision_supprimée
Invoque : suppression_liaison
Procédure : - accéder à la décision id_intro_dialogue
 - id_intro_dialogue du dialogue courant = -1
 - suppression décision.

127 : Objet information- Phase création_info

Objectif : créer un descripteur d'information et le texte associé.

Reçoit : descripteur de l'information

Génère : info_créée
info_annulée

Invoke : création_texte_info
valider_texte_info
insérer_texte_info

Procédure: nb_util = 0
acquérir_valider_support
acquérir_valider_type
acquérir_valider_forme
acquérir_valider_mode_présent

si type = LOCAL
alors : création_texte_info
validation_texte_info

si résultat positif
alors : insérer_texte_info

retour environnement père.

- Phase modification_info

Objectif : modifier texte de l'information

Reçoit : descripteur de l'information

Génère : info_créée
info_annulée

Invoke : extraire_info
éditer_info
validation_texte_info
insérer_texte_info

Procédure: si type = LOCAL
alors : extraire_info
éditer_info
validation_texte_info
si résultat positif
alors : insérer_texte_info

retour environnement père.

- Phase affichage_info

Objectif : afficher selon le mode de présentation associé à l'information, le type d'information spécifié dans le descripteur.

Reçoit : descripteur de l'information

Génère : info_affichée

Invoque : contrôle_cde_affichage_texte
afficher_écran_texte

Procédure: si type de l'information = LOCAL
alors :
 . initialiser variables de contrôle de l'affichage
 . afficher_menu_cde_affichage
 . tant que pas fin_affichage :
 afficher_écran_texte mode_présent
 contrôle_cde_affichage_écran
 . retour environnement père.
si type = CAS
alors : affichage_sel_cas
 cfr Spécifications traitements, phases, 22.1
 p 152

12.8 : Objet format

- Phase création_format

Objectif : créer les attributs de l'objet format_message.

Reçoit : descripteur du format

Génère : format créé

Procédure: nb_util = 0

- acquérir_valider_mode
- si mode = 0
alors : acquérir_valider_forme
 si forme = 's'
 alors :
 . acquérir_noms_des_champs
 . calculer_nb_champs
 . pour_chaque_champ :
 . acquérir_valider_type
 . acquérir_valider_lgr
 . acquérir_valider_importance
 . calculer_col, ligne, en fonction de la longueur du nom et de lgr des champs
- retour environnement père.

- Phase affichage_format

Objectif : afficher le format tel qu'il a été créé

Reçoit : le descripteur d'un format

Génère : format_affiché

Procédure: - afficher_noms_champs
 - calculer les colonnes d'affichage
 - afficher_types_champs
 - afficher_longueurs_champs
 - afficher_importance_champs
 - retour environnement père

12.9 : Objet modèle_connaissance

- Phase création_modèle_conn

Objectif : préparer la création des objets connaissance.

Reçoit : descripteur modèle_conn

Génère : modèle_conn_créé

Procédure: - initialiser les attributs :
 nb_util = nb_formats_util = 0
 - acquérir_valider structure

12.10 : Objet connaissance

- Phase création_connaissance

Objectif : créer la connaissance de référence locale ou globale associée à un format de valeurs dans un modèle de connaissance.

Reçoit : descripteur de connaissance

Génère : connaissance_créée

Procédure: - id_int = 100 x env + id_env
 - nb_util = 0
 - acquérir_valider env_format [0]
 - acquérir_valider id_format [0]
 - acquérir_valider forme
 - acquérir_valider identifiant information utilisée.

- Si forme = 'l' et forme du format déterminé = 'l'
alors :
 - . acquérir_valider fonction de conversion
 - . acquérir_valider commande
 - . calculer nb_cl_mots
 - . retour environnement père.
- Si forme = 'l' et forme du format associé = 's'
alors :
 - si mode du format associé :
 - . = 'o' alors . cpt_classes = nb_classes = 0
 - . = 'm' alors [. afficher info associée
 . acquisition_réponse_marquage
 . retour environnement père
 - . = 'c' alors [. afficher info associée
 . acquérir_réponse_choix multiple
 . retour environnement père
 - . = 'r' alors [. afficher info associée
 . acquérir_réponse_remplissage
 . retour environnement père.
- Si forme = 'g' et forme du format associé = 's'
alors :
 - . acquérir_valider_code
 - . acquérir_valider_param en fonction du code
 - . mise à jour des env_format et id_format utilisés

sinon: message_erreur
- retour environnement père

Règle concernant la connaissance globale :

cfr. Deuxième partie II-21.22 : Spécification des éléments. p 74 et s.

- Phase modification_connaissance

Objectif : si la forme de la connaissance est locale et la forme du format associé est structurée, alors la modification est autorisée au niveau des classes de valeurs, sinon elle est interdite.

Reçoit : descripteur de la connaissance

Génère : connaissance_modifiée
annulation_commande

Procédure: si forme \neq 'l' ou forme du format associé \neq 's'
alors :
 . message_erreur
 . retour environnement père.

- Phase suppression_connaissance

Objectif : supprimer le descripteur de la connaissance
supprimer les classes de valeurs si la connaissance en contient.

Reçoit : descripteur de connaissance

Génère : connaissance_supprimée

Procédure: si forme = 'l'
alors :
 . accéder_classe_valeurs
 . pour les nb_elem valeurs: pres = 0
retour environnement père.

- Phase affichage_connaissance

Objectif : afficher les attributs et valeurs associés à une connaissance.

Reçoit : descripteur de la connaissance

Génère : connaissance_affichée

Procédure: si forme = 'l'
alors:
 si forme du format associé = 'l'
 alors :
 . afficher_commande
 . pour chaque classe de mots clés :
 afficher_classe
 sinon : pour chaque classe de valeurs :
 afficher_entité_classe
 afficher_classe_valeurs

 sinon : décoder et afficher code
 décoder et afficher param
retour environnement père.

12.11 : Objet classe valeurs

- Phase création_classe

Objectif : créer un ensemble de valeurs de même degré de pertinence dans un objet connaissance.

Reçoit : descripteur de connaissance

Génère : classe_créée_enregistrée
classe_annulée.

Procédure: Si nuance = 'c'
alors :
 . id_classe_correcte dans la connaissance locale
 = ident de la classe
 . afficher_format associé
 . tant que pas fin_classe:
 contrôle_création_valeur
si pas annulation
alors :
 . acquérir commentaire
 . insérer valeurs dans le fichier
 . incrémenter nb_classe et cpt_classe
 retour environnement père.

- Phase modification_classe

Objectif : modifier les valeurs d'une classe de valeurs de la connaissance locale d'un objet connaissance.

Reçoit : descripteur de connaissance

Génère : classe_modifiée
classe_annulée

Procédure: - copier en mémoire la classe des valeurs
 - afficher_format associé
 - tant que pas fin_classe :
 contrôle_création_valeur
 - si pas annulation
 alors : insérer valeurs dans le fichier,
 mettre à jour le nombre d'éléments
 - retour environnement père.

- Phase affichage_classe_valeurs

Objectif : afficher écran par écran l'ensemble des valeurs d'une classe.

Reçoit : descripteur de connaissance

Génère : classe affichée

Procédure: - initialiser variables de contrôle d'affichage
- afficher_menu affichage_valeur
- tant que pas fin_affichage :
 . afficher_écran_valeurs
 . contrôle_commande_aff_val
- retour environnement père.

- Phase suppression_classe_valeurs

Objectif : supprimer la description de la classe et l'ensemble de ses valeurs.

Reçoit : descripteur de connaissance

Génère : classe_supprimée

Procédure: - décrémenter nb_classes de la connaissance locale
- compacter les descriptions de classes dans la connaissance locale
- pour les nb_elem valeurs de la classe :
 pres = 0 .

12.12 : Objet modèle_analyse

- Phase création_modèle_analyse

Objectif : créer la séquence des instructions à exécuter pour effectuer l'analyse d'un message.

Reçoit : descripteur de modèle_analyse.

Génère : modèle_analyse_créé.

Procédure: - nb_util = nb_inst = 0
- acquérir_valider env_format [0]
- acquérir_valider id_format [0]
- afficher_menu séquence_instr
- tant que pas fin_modèle_an :
 . acquérir_valider_cde_édition_instr

```

. si cde = insérer
  alors :
    . créer_instruction_analyse
    . incrémenter nb_util de l'objet de l'instruction.
. si cde = détruire
  alors :
    détruire_instruction_analyse.

```

Règles de construction d'une instruction :
 cfr. Spécification des éléments dans
 Deuxième partie 21.22 p 44 ~

- Phase modification_modèle_analyse

Objectif : modifier la séquence des instructions à exécuter pour effectuer l'analyse d'un message.

Reçoit : descripteur de modèle_analyse

Génère : modèle_analyse_modifié

Procédure: - afficher_séquence_instr du modèle
 - la procédure de modification de la séquence est identique à celle de la création des instructions dans la phase création_modèle_analyse.

- Phase affichage_modèle_analyse

Objectif : afficher l'ensemble de la séquence des instructions décodées du modèle d'analyse.

Reçoit : descripteur de modèle_analyse.

Génère : modèle_an_affiché

Procédure: pour chaque instruction :
 . décoder et afficher code
 . décoder et afficher fonction en fonction du code.
 . décoder et afficher param en fonction de code et fonction.

- Phase enregistrer_modèle_analyse

Objectif : recopier la séquence d'instruction du modèle dans le descripteur correspondant.

Reçoit : descripteur modèle_analyse.

Génère : modèle_analyse_enregistré.

Procédure: - pour chaque instruction créée ou modifiée :
 copier instruction dans le descripteur.
 - retour environnement père.

12.13 : Objet règle_lien_dec

- Phase création_lien_dec

Objectif : créer une règle de chaînage entre plusieurs échanges.

Reçoit : descripteur de décision.

Génère : règle_lien_dec_créée
 règle_lien_dec_annulée.

Procédure: - tant que pas fin_condition :
 . acquérir_valider_cde_condition
 si cde = 's'
 alors : créer_expr. relationnelle simple
 si cde = 'sinon' ou ' '
 alors : fin_condition
 - réajuster nb_conditions
 - tant que pas fin_réaction et pas annulation :
 . acquérir_valider_cde_édition_instr.
 si cde = insérer
 alors : . créer_instruction_réac_dec
 . incrémenter nb_util de l'objet de
 l'instruction
 si cde = détruire
 alors : détruire_instruction_réac_dec
 - réajuster nb_réactions.

- Phase modification_règle_lien_dec

Objectif : modifier une règle de chaînage entre échanges

Reçoit : descripteur de décision

Génère : règle_lien_dec modifiée.

Procédure: - recréer conditions
 - afficher séquence_instructions de la réaction
 - procédure de modification des instructions
 est identique à celle utilisée pour créer
 ces instructions.

- Phase affichage_règle_lien_dec

Objectif : afficher, après décodage, les conditions et réactions d'une règle de chaînage entre échanges.

Reçoit : descripteur de décision

Génère : règle_lien_dec affichée

Procédure: - pour chaque condition :
 . afficher 'si'
 . decoder_afficher expression relationnelle simple
 - afficher 'alors'
 - pour chaque instruction de réaction
 . decoder_afficher code
 . decoder_afficher type_obj, en_obj et obj
 - retour environnement père

- Phase suppression_règle_lien_dec

Objectif : supprimer une règle de chaînage d'un objet décision.

Reçoit : descripteur de décision

Génère : règle_lien_dec_supprimée

Procédure: - décrémenter nb_décisions
 - compacter les descriptions des règles de décision.

- Phase enregistrement_lien_dec

Objectif : copier la séquence d'instructions de la réaction de la règle dans le descripteur de décision.

Reçoit : descripteur de décision

Génère : règle_lien_dec_enregistré

Procédure: - pour chaque instruction de la réaction :
 copier instruction dans le descripteur
 - retour environnement père.

12.14 : Objet règle_lien_liaison

- Les phases de création, modification, suppression, affichage et enregistrement sont identiques à celles des objets `règle_lien_dec` quant à leur forme.

- Les différences se situent au niveau des codes.

Cfr. Spécification des éléments : Deuxième partie II, 2.1.2. p. 74 s.s.

13. Phases communes à plusieurs objets- Phase suppression

Objectif : supprimer un objet non utilisé par d'autres objets, ainsi que tous ses sous-objets.

Reçoit : descripteur de l'objet

Génère : objet_supprimé
objet_non_supprimé

Invoque : oper [SUPPRESSION] [sous-objet]

Procédure : si nb_util > 0
 alors : nb_util = nb_util -1
 message erreur
 objet_non_supprimé
 sinon : réajuster fenêtres de l'environnement de
 l'objet,
 pour tous les types de sous-objets de
 l'objet
 pour tous les sous-objets d'un type
 oper [SUPPRESSION] [sous-objet]
 Pres = 0

- Phase affichage

Objectif : afficher en continu tous les sous-objets locaux d'un objet.

Reçoit : descripteur de l'objet

Génère : objet_affiché

Invoque : oper [AFFICHAGE_CONT] [sous-objet]

Procédure : réajuster fenêtres de l'environnement de l'objet
 pour tous les types de sous-objets de l'objet
 pour tous les sous-objets d'un type
 oper [AFFICHAGE_CONT] [sous-objet]

- Phase édition

Objectif : éditer en continu tous les sous-objets locaux d'un objet.

Reçoit : descripteur de l'objet

Génère : objet_édité

- Invoque : édition [sous-objet]

Procédure : réajuster fenêtres de l'environnement de l'objet
 Pour tous les types de sous-objets d'un objet
 pour tous les sous-objets d'un type
 édition [sous-objet.]

- Phase_lister

Objectif : lister tous les objets d'un même critère d'un environnement.

Reçoit : descripteur de l'objet

Génère : liste_objet

Utilise : select_env_crit

Procédure : sélection dans l'environnement courant de l'objet courant, tous les objets dont le critère a été commandé (par défaut: critère de validité), en demandant d'afficher l'ensemble des valeurs des critères de sélection de chacun d'eux.

- Phase_sélection

Objectif : indiquer que l'on veut pouvoir changer d'environnement.

Reçoit : un descripteur d'objet

Génère : Objet_enregistré_cohérent
objet_enregistré_incohérent
annulation
objet_sélectionné

Procédure : - si l'objet sélectionné est l'objet père
 alors: tester si l'objet a été enregistré
 si résultat = non
 alors : enregistrement_objet
 - objet_sélectionné.

- Phase_interruption

Objectif : interrompre la conception d'une leçon, d'un dialogue ou d'un échange pour y revenir ensuite.

Reçoit : descripteur de l'objet

Génère : indicateur d'interruption

Procédure : sauver l'environnement
 indicateur d'interruption.

- Phase enregistrement

Objectif : insérer un objet dans le fichier correspondant après avoir opéré un test syntaxique sur cet objet.

Reçoit : descripteur de l'objet

Génère : objet_enregistré_cohérent
objet_enregistré_incohérent
annulation_commande.

Invoke : test_cohérence [sous-objet]

Procédure : si test_cohérence [type-objet] de l'objet < 0
 alors : demande_de_contrôle
 si réponse = non
 alors : objet_enregistré_incohérent
 incrémenter compteur d'objet dans
 l'environnement de validité,
 sinon : annulation_commande
 sinon : - insérer objet dans le fichier corres-
 pondant
 - incrémenter compteur d'objet dans l'en-
 vironnement de validité
 - objet enregistré cohérent.

- Phase modification

Objectif : préparer la modification des sous-objets d'un objet.

Reçoit : descripteur de l'objet.

Génère : /

Invoke : /

Procédure : /

- Phase get_commande_op

Objectif : acquérir et valider la commande opération

Reçoit : fin_commande

Génère : cde_op_valide

Invoke : test_command_op

Procédure : afficher_menu_opération
 acquérir_valider_opération
 si résultat ≠ annulation ou fin_travail
 alors : mise à jour variables globales
 afficher_commande

- Phase get_command_ob

Objectif : acquérir et valider la commande objet
Reçoit : cde_op_valide
Génère : cde_obj_valide
 annulation
Invoke : test_command_obj
Procédure : afficher menu_objets de l'environnement
 acquérir_valider_objet
si résultat \neq annulation
alors : mise à jour variables globales
 afficher_commande.

- Phase get_command_sélection

Objectif : acquérir et valider le critère de sélection de
 l'objet et sa valeur;
 si plusieurs objets répondent au même critère,
 la liste de ces objets est affichée.
Reçoit : cde_obj_valide
Génère : cde_select_valide
 annulation
Invoke : test_command_select
 test_attribut_sel
Procédure : si l'objet possède des critères de sélection
alors :
 afficher_menu_sélection
 afficher_entité_sélection
si l'opération courante = CREATION
 afficher les valeurs par défaut des critères
tant que pas fin_sélection
 acquérir_valider_critère_sel
si résultat \neq ANNULATION et \neq fin_sélection
alors :
 acquérir_valider_valeur_sélection
si résultat \neq ANNULATION
alors : cde_select_valide.

2. APPLICATION 2

21. Modules de coordination

21.1 init_environ

21.2 gestion_dialogue

21.3 clôture_environnement

21.1 : Init_environt

Identique à la phase correspondante de l'application 1.

21.2 : Gestion_dialogue

Objectif : coordonner l'enchaînement des dialogues et échanges

Reçoit : environnement initialisé

Génère : fin_dialogue

Invoke : sélection_leçon
exécuter_liaison
exécuter_décision
exécuter_instruction_échange

Procédure: - tant que pas fin_exécution :
 . sélection_leçon à exécuter
 . activer leçon
 . trace_leçon
 . exécuter_liaison id_liaison_intro de la leçon courante
 . tant que pas terminer_leçon :
 . activer_dialogue
 . trace_dialogue
 . exécuter_décision id_décision_intro du dialogue courant
 . tant que pas terminer_dialogue :
 . activer_échange
 . trace_échange
 . exécuter_échange
 . exécuter_décision décision courante
 . si aucune condition vérifiée :
 alors : terminer_dialogue
 message_erreur.
 . exécuter_liaison liaison courante
 . si aucune condition vérifiée
 alors : terminer_leçon
 message_erreur
 - demande de gestion_dialogue :
 si cde = 'a' alors: afficher_trace
 'e' éditer_trace
 't' terminer_exécution
 'c' continuer_exécution
 'd' détruire_trace.

21.3 : Clôture_environnement

Objectif : clôturer l'ensemble des fichiers utilisés.

Reçoit : fin_exécution

Génère : fin_travail
trace_édité

Procédure: clôturer des fichiers
compacter_fichiers_traces.

22. Phases associées aux objets

22.1 : Objet cas

- Phase affichage_sel_cas

Objectif : afficher une partie standard déterminée d'un dossier ou une liste de symptômes déterminés par certains critères issus du dossier.

Reçoit : descripteur d'information avec type = cas

Génère : sous_partie_dossier_affiché

Procédure: .créer_fichier_commande avec commande =
type du cas correspondant, A, no_dos, commande
du descripteur d'information.

.exécuter en sous-processus le programme
gestion_cas.

.extraire les informations du fichier renvoyé.

.si fichier ≠ vide

alors : les informations concernent des
symptômes;
afficher_liste_symptômes.

22.2 : Objet échange

- Phase exécuter_échange

Objectif : exécuter la suite des instructions de la
séquence de l'échange interactif.

Reçoit : descripteur d'échange

Génère : info_affichée
message_reçu
message_analysé.

Procédure : - accès au modèle de connaissance associé
 - pour chaque instruction de l'échange :
 exécuter_instruction_echange.

22.3 : Objet liaison

- Phase exécuter_liaison -----

Objectif : évaluer les règles de chaînage en fonction
 des renseignements de l'environnement.

Reçoit : descripteur de liaison.

Génère : règle_instanciée
 aucune_règle_instanciée
 dialogue_courant
 liaison_courante

Invogue : évaluation_condition_elem_li

Procédure : - pour chaque règle de décision et
 tant que évaluation_condition négative
 pour chaque condition d'une règle et
 tant que évaluation positive
 évaluation_condition_elem_li
 - si aucune règle instanciée
 alors : message erreur
 sinon : pour chaque instruction de la règle
 instanciée:
 exécuter_instruction_li

22.4 : Objet décision

- Phase exécution_décision -----

Cette phase est identique à la phase exécution_dédicion
 quant à sa forme.

22.5 : Objet information

- Phase affichage_information -----

Cette phase a été spécifiée dans l'application 1
 cfr. 12.7 : Objet information.

22.6 : Objet format_message

- Phase affichage_entité_format

Objectif : afficher le nom du format ainsi que celui des différents champs s'il en existe.

Reçoit : descripteur de format.

Génère : format_affiché

- Phase acquisition_message_marquage

Objectif : acquérir les réponses en mode marquage, soulignements de mots dans un texte, permettre de modifier ensuite la liste extraite et de pouvoir continuer à souligner d'autres mots, de commander une consultation ou requête.

Reçoit : descripteur de l'information

Génère : liste_mots_marques

Procédure: - affichage_information

- tant que pas fin_message

. acquérir_valider_cde_marquage

si cde = '>',

alors: déplacer curseur vers la droite

si cde = 'v'

alors: déplacer curseur vers le bas

si cde = 'm'

alors: changer de mode:

mode_non_marq → mode_marq OU

mode_marq → mode_non_marq

si cde = 'l'

alors: afficher_modifier liste des réponses extraites.

si cde = 'f'

alors: fin_message

si cde = 'r'

alors: acquérir_exécuter_requête
afficher_information

si cde = 'c'

alors : acquérir_valider_consult
afficher_information.

- Phase acquisition_message_remplissage

Objectif : acquérir les réponses en mode remplissage de zones libres d'un texte et permettre de modifier la valeur de certaines zones, de commander une consultation ou requête.

Reçoit : descripteur d'information

Génère : liste_mots

Procédure: - affichage_information

- tant que pas fin_message :

 . acquérir_valider_cde_remplissage

si cde = '>'

alors: passer à la zone suivante

si cde = 'v'

alors: acquérir_valider_zone

si cde = '<'

alors : passer à la zone précédente

si cde = 'f'

alors : fin_message

si cde = 'c'

alors : acquérir_exécuter_consult

 afficher_information

si cde = 'r'

alors: acquérir_valider_requête

 afficher_information.

- Phase acquisition_message_choix_multiple

Objectif : acquérir une suite de nombres déterminant les choix de réponses et permettre de commander une consultation ou requête.

Reçoit : descripteur de l'information

Génère : liste_nombres

Procédure: tant que pas fin_message :

 . acquérir_valider_cde_choix_mult

si cde = 'v'

alors : acquérir_valider_nombre

si cde = 'r'

alors : acquérir_exécuter_requête

 afficher_information

si cde = 'c'

alors : acquérir_exécuter_requête

 afficher_information

si cde = 'f'

alors : fin_message

- Phase acquisition_valeurs

Objectif : - acquérir une liste de valeurs, chacune étant spécifiée par son format,
- permettre de commander une consultation ou requête.

Reçoit : -

Génère : liste de valeurs classées par format.

Procédure: - tant que pas fin_message :
 . acquérir_valider_cde
 si cde = 'v'
 alors
 si un seul format autorisé
 alors
 si forme = '1'
 alors : acquisition_ligne_rep
 sinon : acquérir_valider_syntaxe_valeur
 sinon : afficher_menu formats autorisés
 acquérir_valider_cde
 acquérir_valider_syntaxe_valeur
 si cde = 'r'
 alors : acquérir_exécuter_requête_autorisée
 afficher_réponses
 si cde = 'c'
 alors : acquérir_exécuter_consult_autorisée
 afficher réponses
 si cde = 'f'
 alors : fin_message.

3. APPLICATION 3

Module_de_coordination

- Phase init_envir

Objectif : ouvrir les fichiers_trace

Reçoit : demande_de_suivi

Génère : fichiers_ouverts

- Phase gestion_cde_stat

Objectif : acquérir, valider et exécuter les demandes de calcul, d'affichage ou d'édition de statistiques.

Reçoit : fichiers_trace_ouverts et fin_commande

Génère : fin_commande
fin_travail

Procédure: tant que pas fin_travail :

. acquérir_valider_cde de traitement

. si pas fin_travail

alors : acquérir_valider commande de statistiques
exécuter cde de traitement.

Règle: la syntaxe des statistiques à commander reste à spécifier.

- Phase clôture_envir

Objectif : fermer les fichiers_trace et d'édition

Reçoit : fin_travail

Génère : fin_suivi.

22.24 : Fonctions

CONVENTIONS :

- Seules les fonctions importantes invoquées par les phases seront spécifiées.
- Elles seront regroupées par application

1. APPLICATION 1

- 1.1. Fonctions associées aux phases de coordination
- 1.2. Fonctions associées aux phases concernant les objets.

1.1. Fonctions associées aux phases de coordination

- Fonction test_commande_op

Objectif : tester une commande d'opération

Reçoit : cde

Génère : cde_valide
fin
annulation
cde_invalide

Procédure: si cde = opération de gestion
alors : cde valide
sinon :
 si cde = 'f'
 alors : fin
 si cde = '?'
 alors : afficher écran d'aide
 si code = '>'
 alors : afficher_suite du menu
 si cde = '!'
 alors : annulation

- Fonction test_command_ob

Objectif : tester une commande d'objet dans l'environnement courant.

Reçoit : cde

Génère : objet_valide
annulation

Procédure: si cde = objet valide dans l'environnement
alors : objet_valide
sinon :
 si cde = '!'
 alors : annulation
 si cde = '?'
 alors : afficher écran aide
 si cde = '>'
 alors : afficher suite écran menu
 si cde = '^'
 alors : environnement = environnement père.

- Fonction test_command_select

Objectif : tester le critère de sélection d'un objet du type courant.

Reçoit : cde

Génère : critère_valide
 annulation

Procédure: si cde = critère de sélection possible de l'objet courant,
alors : critère valide
sinon :
 si cde = '?'
 alors : afficher_écran aide
 si cde = '!'
 alors : annulation
 si cde = '>'
 alors : afficher_suite du menu
 si cde = '1'
 alors : lister tous les objets du type courant de l'environnement courant.
 si cde = 'f'
 alors : fin_select.

- Fontion test_attribut_sel

Objectif : tester la valeur du critère de sélection d'un objet du type courant.

Reçoit : commande

Génère : val_crit_valide
 val_crit_multiple
 val_crit_absent
 annulation


```

Procédure: si commande = '!'
              alors: annulation
              sinon: test_syntaxique du contenu de commande par
                      rapport au type élémentaire associé au
                      critère courant.

              si résultat négatif
              alors : message_erreur
              sinon :

                  si opération = création ou modification
                  alors : test_sémantique du contenu de commande
                          associé au critère courant.

                  si résultat négatif
                  alors: message_erreur
                  sinon : val_crit_valide

              sinon : sélectionner tous les objets de
                      l'environnement répondant au critère

                  si : il en existe plusieurs
                  alors: val_crit_multiple

                  si: il n'en existe aucun
                  alors : val_crit_absent
                  sinon: val_crit_valide.

```

1.2. Fonctions associées aux phases concernant les objets

- Fonction test_cohérence_leçon

Objectif : tester s'il existe des liaisons ou dialogues
incohérents et s'il existe une première liaison.

Reçoit : description leçon

Génère : leçon cohérente
leçon_incohérente

Procédure: si nb_liaisons_incohérentes > 0 OU
 nb_dialogues_incohérents > 0 OU
 id_liaison_intro = -1
 alors : leçon incohérente
 sinon : leçon cohérente.

- Fonction test_cohérence_dialogue

Objectif : tester s'il existe des décisions ou échanges incohérents et s'il existe une première décision.

Reçoit : descripteur de dialogue

Génère : dialogue_cohérent
dialogue_incohérent

Procédure: si nb_décisions_incohérentes > 0 OU
nb_échanges_incohérents > 0 OU
id_décision_intro = -1
alors : dialogue_incohérent.
sinon : dialogue cohérent

- Fonction test_cohérence_liaison

Objectif : cfr. Spécification des éléments
Deuxième partie II-21.22 p. 74 r.s.

Reçoit : descripteur liaison

Génère : liaison cohérente
liaison incohérente

Procédure: règle 1 :

combinaisons_de_code_interdites

a	-	-	-	-
t	-	o	-	n
e	o	-	n	-
r	o	-	n	-
f	-	o	o	n

avec :

o = oui

n = non

- = oui ou non

De plus, seule l'instruction de code 'a' peut être multiple.

règle 2

Les dialogues et liaisons utilisés doivent exister dans les environnements de validité accessibles.

- Fonction test_cohérence_décision

Objectif : cfr. Spécification des éléments
Deuxième partie II-21.22 p. 74 ss.

Reçoit : descripteur de décision

Génère : décision_cohérente
décision_incohérente

Procédure: règle 1 :

combinaisons_interdites

a	-	-	-	-	-	-
t	-	-	-	o	-	n
e	o	o	n	-	n	-
r	o	n	o	-	n	-
c	n	o	o	-	n	-
p	-	-	-	o	o	n

De plus, seule l'instruction de code 'a' peut être multiple.

Règle 2 :

Les échanges et décisions utilisés doivent exister dans les environnements de validité accessibles.

- Fonction test_cohérence_échange

Objectif : cfr. Spécification des éléments
Deuxième partie II-21.22 p. 74 ss.

Reçoit : descripteur_échange

Génère : échange_cohérent
échange_incohérent

Procédure: Règle :

les combinaisons_interdites :

t	-	-
r	n	o
a	o	n

seule, l'instruction de code 'a' peut être multiple

- Fonctions : test_cohérence_info
 test_cohérence_format
 test_cohérence_modèle_connaissance
 test_cohérence_connaissance

 donnent systématiquement un résultat positif, tout étant vérifié en cours de création ou modification.

- Fonction validation_texte_info

Objectif : opérer un texte syntaxique sur le texte d'une information provenant, soit d'un fichier édité, soit du terminal, dans le cas où celle-ci est de type LOCAL, et permettre la correction du texte en cas d'erreur.

Reçoit : descripteur d'information
 indicateur = 0 si texte est dans un fichier édité
 = 1 si texte est introduit directement au terminal.

Génère : texte_valide
 texte_annulé

Procédure: analyser_texte_info
 si résultat négatif
 alors : demande_de_correction
 si cde = 'e'
 alors : afficher_fichier_erreur
 si cde = 'a'
 alors : abandonner
 si cde = 'c'
 alors : éditer_fichier_à corriger

Règle d'analyse d'un texte:

si type information = LOCAL

alors :

 , si mode_présent = 's'

alors : < ligne > : = Ø | < chaîne_car >

 < chaîne_car > : = Ø caract < chaîne_car >


```

. si mode_présent = 'c'
  alors : < ligne > := g | n | i | s | c |
                                1 < no > | c < no > | blanc < chaîne_car >
    < no > := nombre n
  avec : g = gras
          n = normal
          i = inverse
          s = souligné
          c = clignotant
          l = ligne à sauter
          c = colonne

. si forme = 'r'
  alors :- < chaîne_car > := ∅ | caract < chaîne_car >
    | < chaîne_car > [ < type > < caract de remplissage >
    < chaîne_caractère >
    - < type > := type élémentaire
    - < caractère de remplissage > = ∅
    - < caractère de remplissage >
  sinon : < chaîne_car > := ∅ | caract < chaîne_car >

```

- Fonction création_texte_info

Objectif : permettre la création d'un texte d'information à l'aide de l'éditeur du VAX.

Reçoit : descripteur d'information

Génère : fichier édité
demande_de_création_au_terminal

Procédure : acquérir_valider_demande_éditeur

si réponse = créé

alors : éditer_info
fichier_édité

Sinon : demande_de_création_au_terminal

2. APPLICATION 2

Les fonctions spécifiées sont les fonctions d'exécution d'instructions invoquées par les phases de l'application 2

- Fonction exécuter_instruction_dec

Objectif : exécution d'une instruction

Reçoit : instruction_réac_dec

Génère : objet_affiché
échange_courant
décision_courante

Procédure: règles :

code	type_obj	env_obj	obj	Fonction
'a'	EVALUATION			Afficher l'ensemble des renseignements résultant de l'analyse situés dans les environnements env_dialogue et env_échange.
	OBJET	env	id	Afficher un objet particulier de l'environnement env_échange ou env_dialogue
	REPONSE CORRECTE	env	id	Afficher réponse correcte.
	INFORMATION	env	id	Affichage_information
	COMMENTAIRES			Afficher_commentaires générés par l'analyse.
'r' ou	ECHANGE	env	id	Réinitialiser les compteurs.
'e'	ECHANGE	env	id	Mettre à jour échange courant
'c'	ECHANGE	env	id	Mettre à jour échange courant.
'p'	DECISION	env	id	Mettre à jour décision courante
't'				Terminer_dialogue

- Fonction exécuter_instruction_li

Objectif : exécuter une instruction de réaction d'une règle de chaînage d'une liaison.

Reçoit : instruction_reac_li

Génère : objet_affiché
dialogue_courant
liaison_courante

Procédure: règles :

code	type_obj	env_obj	obj	Fonction
'a'	EVALUATION			Afficher l'ensemble des renseignements des structures auxilliaires env_dialogue et env_leçon.
'a'	OBJET	env	id	Afficher la valeur d'un objet id de l'environnement env (env = env_dialogue ou env_leçon.
'a'	INFORMATION	env	id	Accéder_information Afficher_information
'r' ou	DIALOGUE	env	id	Réinitialiser les compteurs.
'e'	DIALOGUE	env	id	Mettre à jour dialogue courant.
'f'	LIAISON	env	id	Mettre à jour liaison courante.
't'				Terminer_leçon

- Fonction exécution_instruction_échange

Objectif : exécuter instruction d'un échange.

Reçoit : instruction_échange

Génère : info_affiché
message_reçu
message_analysé.

Procédure: - si code = 'e'
 alors : . accès à l'information
 . afficher information

si code = 'r'
 alors :
 si un seul format autorisé
 alors :
 . si mode_format = 'm'
 alors : acquisition_réponse_marquage

 . si mode_format = 'r'
 alors : acquisition_réponse_remplissage

 . si mode_format = 'c'
 alors : acquisition_réponse_choix_multiple.
 sinon : acquisition_valeurs.

sinon : acquisition_valeurs

- si code = 'a'
 alors :
 pour chaque format autorisé
 pour chaque instruction du modèle_analyse
 associé :
 exécuter_instruction modèle_analyse.

- Fonction exécution_instruction_modèle_analyse

Objectif : exécuter une instruction d'analyse sur la liste
 des valeurs acquises.

Reçoit : liste_valeurs
 descripteur_modèle_analyse.

Génère : paramètres d'évaluation
 cfr. Annexe Structures auxiliaires env_échange.
 trace_valeurs

Procédure:

Règle : instruction

code	fonction	param	fonction
'c'	COMP_DIFF COMP_CLASSE CMP_LIBRE	cfr. Spécification éléments IIe partie II-21.2	Exécuter comp_diff Exécuter comp_classe Exécuter cmp_libre
'm'	FREQUENCE DEGRE ELECTRE DIAGNOSTIC_DIFF PATHOGENESIE MOTS_CLES	Non encore spécifiés	Mod_fréquence Mod_degré Mod_électre
'e'	EVAL_STD EVAL_CERT		Exécuter évaluation standard Calculer_évaluation_ avec_ceritude

Règle : chargement paramètres.

Type = CORRECTE : si forme connaissance associée = LOCALE
alors :
 copier dans la zone standard des valeurs correcte, en mémoire, la classe des valeurs correctes dépouillée des synonymes.

si forme connaissance associée = GLOBALE
alors :
 exécuter_modèle_global
 copier résultat dans zone standard.

Type = APPRENANT: les valeurs se trouvent dans la zone standard des valeurs venant de l'apprenant.

Type = GENEREE : les valeurs ont été chargées dans la zone standard réservée aux réponses générées par un modèle sur base des paramètres donnés par l'apprenant.

22.3 Niveau 'UTILISE' 1

Primitives fonctionnelles

Fonctions :

- ```

. copier [type] (destination, source, longueur)

0 : CARACTERES : cpier_s
1 : ENTIER : cpier_n
2 : REEL : cpier_r
3 : SYPTOME : cpier_sympt
4 : REMEDE : cpier_s
5 : MALADIE : cpier_s

. cmp [type] (zone à comparer, zone de référence, longueur)

0 : CARACTERES : cmp_s
1 : ENTIER : cmp_n
2 : REEL : cmp_r
3 : SYMPTOME : cmp_sympt
4 : REMEDE : cmp_s
5 : MALADIE : cmp_s

. test syntaxique [type] (zone de caractères à tester)

0 : CARACTERES : test_alpha
1 : ENTIER : test_num
2 : REEL : test_réel
3 : SYMPTOME : -
4 : REMEDE : test_remède
5 : MALADIE : test_maladie

. recherche (adresse tableau
 nombre d'éléments du tableau
 taille d'un élément de tableau
 adresse relative dans un élément de l'objet
 à rechercher
 type de l'objet à rechercher
 relation à vérifier
 adresse de la zone contenant la valeur à
 rechercher.)

```

La liste des fonctions n'est pas nécessairement exhaustive.



## 22.4 Niveau 'UTILISE' 2

22.41 Primitives\_d'acquisition\_validationFonctions :1) Primitives\_d'acquisition\_

acquisition\_cde (ligne, colonne, nombre maximum de caractères,  
test à effectuer sur la commande)

acquisition\_ligne (ligne, colonne, nombre maximum de caractères  
à acquérir,  
test à effectuer sur la ligne de commande)

2) Primitives\_de\_test\_sémantique

ident\_obligatoire (commande)

non\_modifiable (commande)

obligatoire (commande)

3) Primitives\_de\_test\_des\_valeurs\_de\_champs

| <u>NOM</u>         | <u>ENTITE CORRESPONDANTE</u> | <u>ATTRIBUT CORRESPONDANT</u> |
|--------------------|------------------------------|-------------------------------|
| tes_support        | information                  | support                       |
| test_type_info     | information                  | type                          |
| test_forme_info    | information                  | forme                         |
| test_mode_info     | information                  | mode_présent                  |
| test_type_obj_cas  | info_cas                     | type_obj                      |
| test_mode_fmt      | format                       | mode                          |
| test_forme_fmt     | format                       | forme                         |
| test_forme_conn    | connaissance                 | forme                         |
| test_conv_lib      | connaissance locale_libre    | conversion                    |
| test_nuance_classe | connaissance locale_classe   | nuance                        |
| test_code_glb      | connaissance globale         | code                          |
| test_code_inst_an  | modèle_analyse               | code                          |
| test_fct_an        | modèle_analyse               | fonction                      |
| test_code_éch      | échange instruction          | code                          |
| test_réel          | décision                     | rel                           |
| test_type_expr_cd  | ou                           | type_expr                     |
| test_code_réac     | liaison                      | code                          |
| test_type_obj_réac | réaction                     | type_obj                      |
| test_type_elem     |                              |                               |

Remarque : en ce qui concerne les valeurs possibles des attributs:  
cfr. Deuxième partie IL2122 p. 74 p. 2.

La liste des fonctions n'est pas nécessairement exhaustive.

## 22.42 Primitives gestion fichiers

## Fonctions :

## 1. Primitives d'accès fichier

### 1.1. Initialisation des fenêtres de validité des sous-objets d'un environnement.

- init\_wnd\_env (type d'environnement  
                  identifiant de l'environnement)
- wnd\_env (type d'environnement,  
          identifiant de l'environnement  
          type du sous-objet particulier)
- search\_key\_env (type d'environnement  
                  identifiant de l'environnement).

### 1.2. Primitives de recherche dichotomique d'un élément dans un environnement.

- ```
- dich_env_key    (identifiant fichier, valeur de clé)
- dich_env_out    (identifiant fichier,
                  type de l'objet cherché,
                  adresse relative dans un record de l'objet
                  cherché,
                  adresse de la zone contenant la valeur
                  de l'objet cherché).
```

1.3. Primitives de recherche séquentielle d'un élément dans un environnement.

- seq_env_key (identifiant du fichier, valeur de clé)
- seq_env_crit (identifiant du fichier,
type de l'objet cherché,
adresse relative dans un record de l'objet
cherché,
adresse de la zone contenant la valeur
de l'objet cherché).

1.4. Primitives de sélection
de tous les éléments répondant à une condition dans un environnement.

- select_env_key (identifiant fichier, valeur de clé,
action à effectuer sur les éléments
trouvés)
- select_env_crit (identifiant fichier,
type des objets cherchés,
adresse relative dans un record des
objets cherchés,
adresse de la zone contenant la
valeur des objets cherchés,
action à effectuer sur les objets trouvés

2. Primitives_de_compactage_et_décalage_

- 2.1. compacter (identifiant du fichier)
- 2.2. décaler (identifiant du fichier)

3. Manipulation_de_valeurs_de_clé

- 3.1. clé_env (type d'environnement,
identifiant de l'environnement)
- 3.2. val_clé (type d'environnement,
identifiant de l'environnement,
zone réceptrice des valeurs)
- 3.3. cmp_clé_s (adresse de l'élément à comparer,
valeur de la clé).

4. Primitive_de_suppression_de_l'élément_courant

supprimer (identifiant du fichier)

5. Attribution_d'un_numéro_dans_l'environnement_courant_à_un_nouvel_élément

attribuer_no (identifiant du fichier).

6. Conversion_d'un_fichier_RMS_en_fichier_flux

convertir_info (identifiant du fichier à convertir
identifiant du fichier de sortie)

7. Extraire_information_du_fichier_local_info.dat

extraire_info (adresse de début de l'information à
extraire
nombre de bytes de l'information à
extraire
identifiant du fichier de sortie).

La liste des fonctions n'est pas nécessairement exhaustive.

22.5 Niveau 'UTILISE' 3

22.51 Primitives d'entrée-sortie

Fonctions : `opn_fle` (`idn`, `mod`)
 `cls_fle` (`idn`, `mod`)
 `idx_inc` (`idn`, `typ`, `mod`)
 `adj` (`idn`, `typ`, `unt`, `dsp`, `mod`)

Contraintes d'utilisation

- Les manipulations se font essentiellement sur des records logiques d'un fichier. Ne pas mélanger des primitives d'action sur des records et sur des bytes pour un même fichier.
- Les primitives `opn_fle`, `adj` et `idx_inc` réajustent, à chacune de leur exécution, le nombre total de records et de bytes existant dans le fichier.
- Toute écriture est faite en fin de fichier. Il est conseillé d'utiliser alors la primitive `idx_inc` (`idn_RCD_OUT`).
- Dès que les opérations d'écriture ou de mise à jour sont effectuées sur un fichier, il est conseillé d'utiliser systématiquement le `mod_I_O` pour les primitives d'ajustement et d'incrémentation d'index manipulant des objets à l'intérieur du fichier, et d'utiliser la primitive `cls` (`idn`, `OUT`) pour fermer le fichier, afin de sauver le dernier buffer.

Utilise :

les fonctions : `_opn_fle` (`idn`, `mod`)
 `_cls_fle` (`idn`)
 `_adj` (`idn`, `typ`, `unt`, `dsp`, `mod`)
 `_idx_inc` (`idn`, `typ`, `mod`)

de la librairie des entrées-sorties écrite par l'U.I.H.N. computer team".

22.52 Primitives gestion-écran

Fonctions

Afficher [type] (ligne, colonne,
longueur de la zone à afficher,
zone, mode d'affichage)

0 : CARACTERES	: afficher_s
1 : ENTIER	: afficher_n
2 : REEL	: afficher_r
3 : SYPTOME	: afficher_symp
4 : REMEDE	: afficher_s
5 : MALADIE	: afficher_s

afficher_menu (identifiant menu)
afficher_environnement ()

error (message à afficher,
nombre de coups de sonnette)

effacer_zone (ligne, colonne, longueur)
clear_screen ()

lec_term (ligne, colonne, longueur maximum à lire)

Utilise : les fonctions

_normal (), _bold (), _reverse (), _flash ()
_underscore (), _bell (n), _delete (), _locate (l,c),
_région (l1, l2), _clear (mod, dir, ligne, lgr, col).

de la librairie des primitives écran écrites par
l' "U.I.H.N. computer team".

La liste des fonctions n'est pas nécessairement exhaustive.

II. 3- CHOIX D'IMPLEMENTATION

II. 31 - Choix du langage de programmation :

- Le langage C a été choisi pour l'écriture des outils du système étant donné que :
 - les interactions continues avec le système d'exploitation de la B.D.H. proposées :
 - . lors de la conception des objets connaissance, par l'éditeur des objets
 - . lors de l'acquisition des messages de l'apprenant par l'outil d'exécution des leçons,
 - certaines des procédures du système navigationnel des répertoires et d'aide à la décision ont été intégrés, après décomposition ou modification, dans le système d'aide à l'enseignement.
- Le langage COBOL sera nécessaire pour traiter les commandes agissant sur des cas de dossiers réels (dans la base des dossiers centralisés), ou fictifs (dans la base des cas fictifs) étant donné que :
 - le système d'exploitation de la base des dossiers est écrit dans ce langage,
 - certains de ses modules pourront être utilisés, avec quelques modifications (nom des fichiers, des records, ...) pour construire et gérer les cas fictifs.

II- 32. Choix d'une organisation de fichiers.

1. Choix de l'organisation.

- Etant donné la structure des objets du système en termes de leur environnement de validité, il est nécessaire que les fichiers correspondants soient triés constamment sur l'identifiant de cet environnement.
- Etant donné les accès très fréquents aux fichiers, le recours à des procédures de gestion de fichiers rédigées en d'autres langages (permettant la manipulation de fichiers RMS) aurait été trop lourd.
- L'organisation des fichiers est séquentielle, les procédures de gestion de fichiers étant rédigées en langage C.
- La découpe en fichiers se superpose à la découpe en objets pédagogiques.
- Le fichier valeurs a été introduit étant donné que le nombre de valeurs possibles pour une classe de valeurs peut varier de 1 à l'infini.

2. Description des fichiers.

Chaque fichier est décrit par :

- ses attributs :

le signe ◁ indiquant qu'un attribut donné peut être multiple

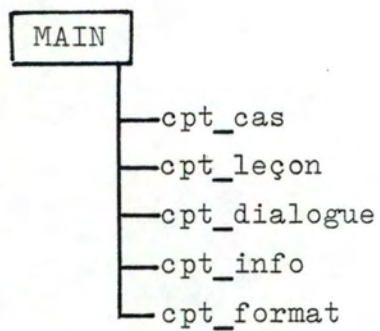
- ses modes d'accès :

clé primaire : représente les attributs identifiant un article du fichier et servant de clé de tri.

clés secondaires : représentent les attributs d'un article sur base desquels la sélection de l'objet correspondant peut être effectuée.

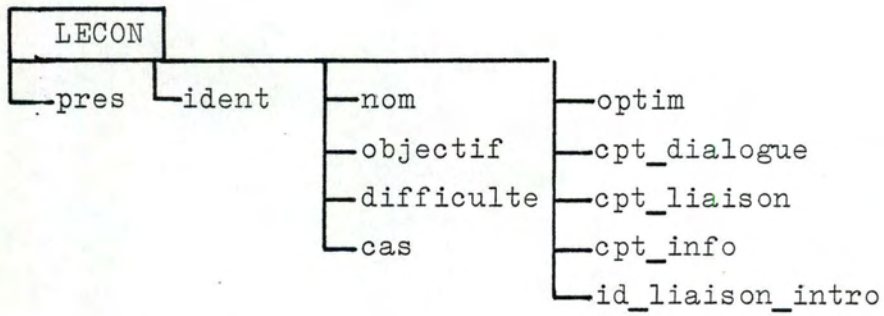
- Liste des fichiers :

. main	. trace_dialogue
. leçon	. trace_échange
. liaison	. trace_valeurs
. dialogue	
. décision	
. échange	
. information	
. format_mess	
. modèle_connaissance	
. connaissance	
. modèle_analyse	
. valeurs	
. cas	
. trace_leçon	



clé primaire : -

clés secondaires : -



clés

primaire id

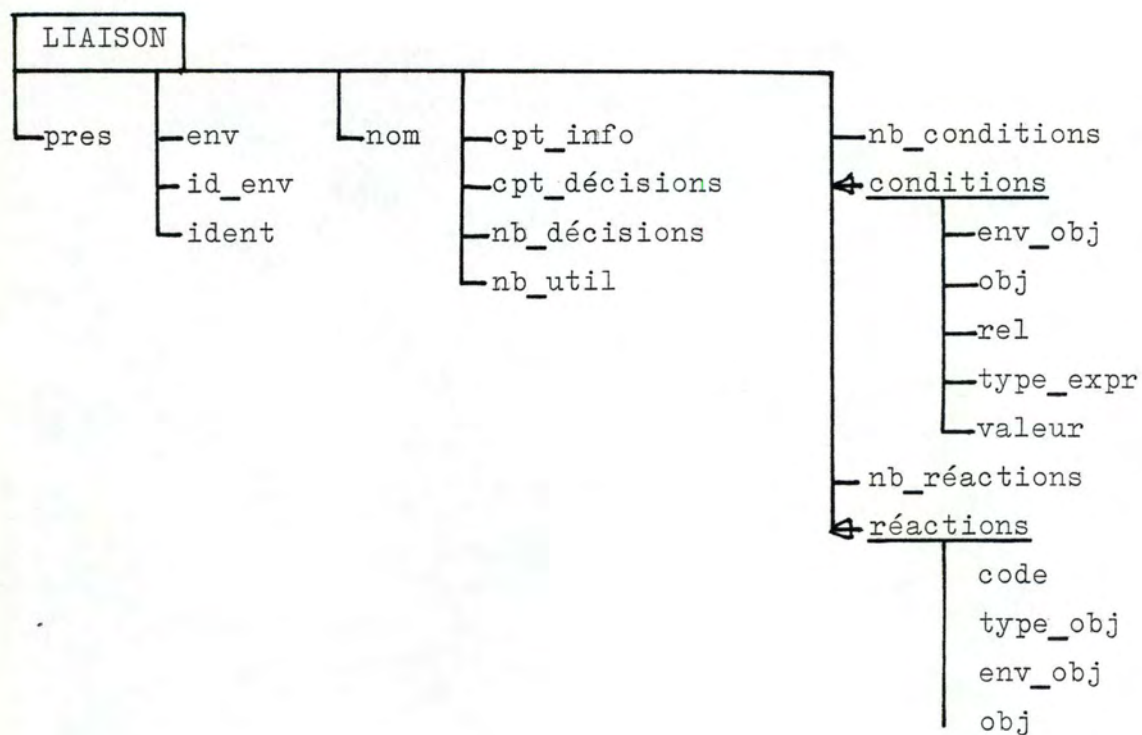
secondaires

identifiante: nom

multiples: objectif

difficulté

cas

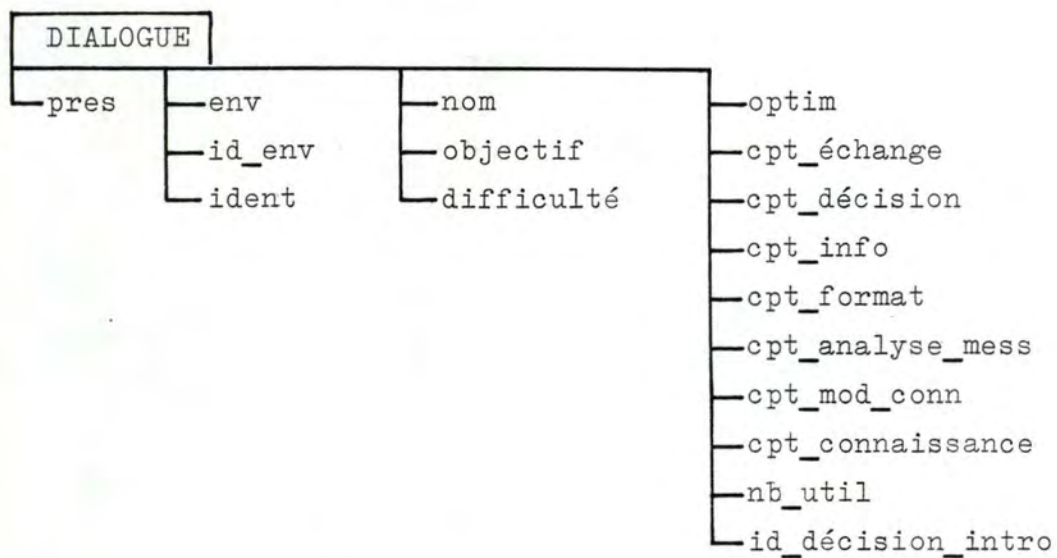


clés

primaire env, id_env, ident

secondaire

identifiante env, id_env, nom



clés

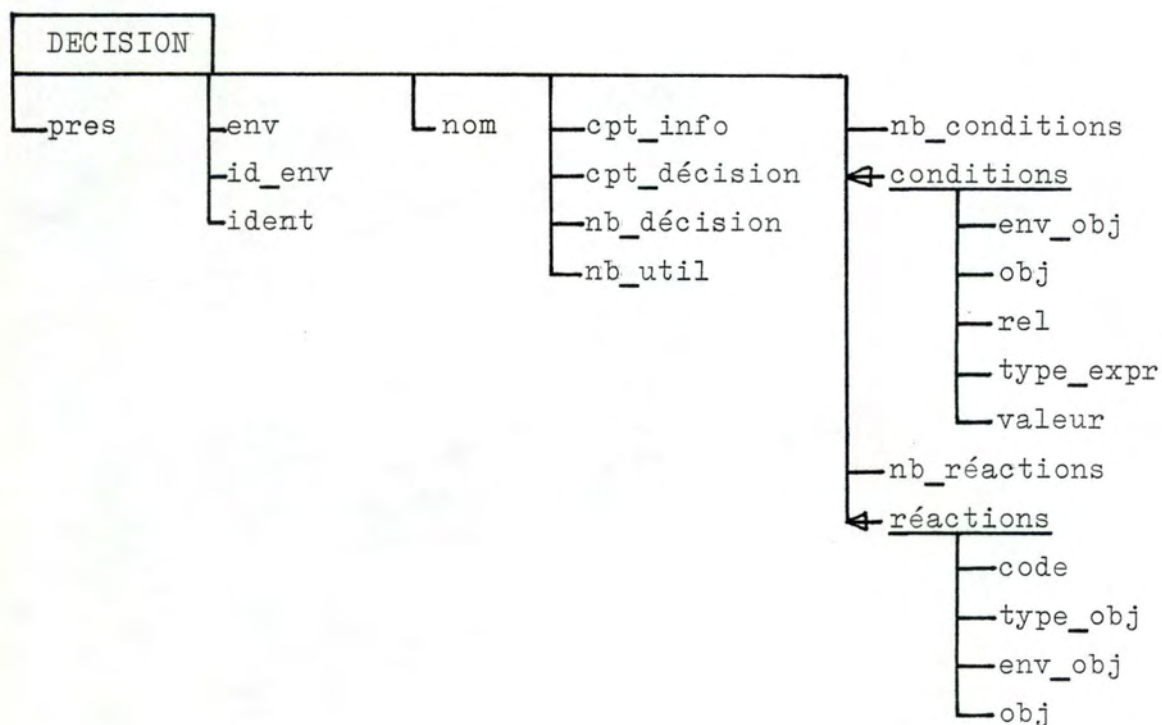
primaire env, id_env, ident

secondaires

identifiante env, id_env, nom

multiples env, id_env, objectif

env, id_env, difficulté

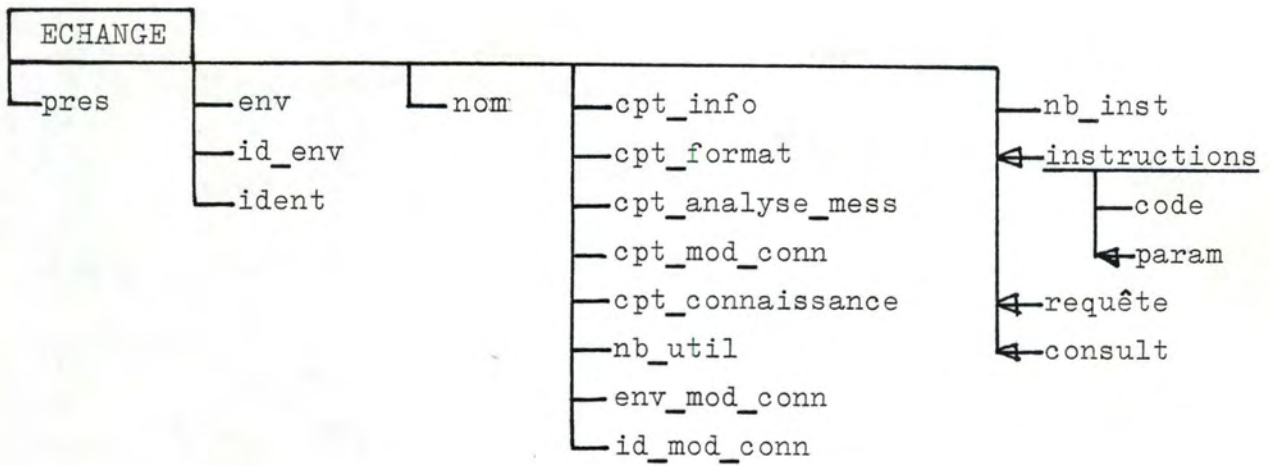


clés

primaire : env, id_env, ident

secondaire: env, id_env, nom

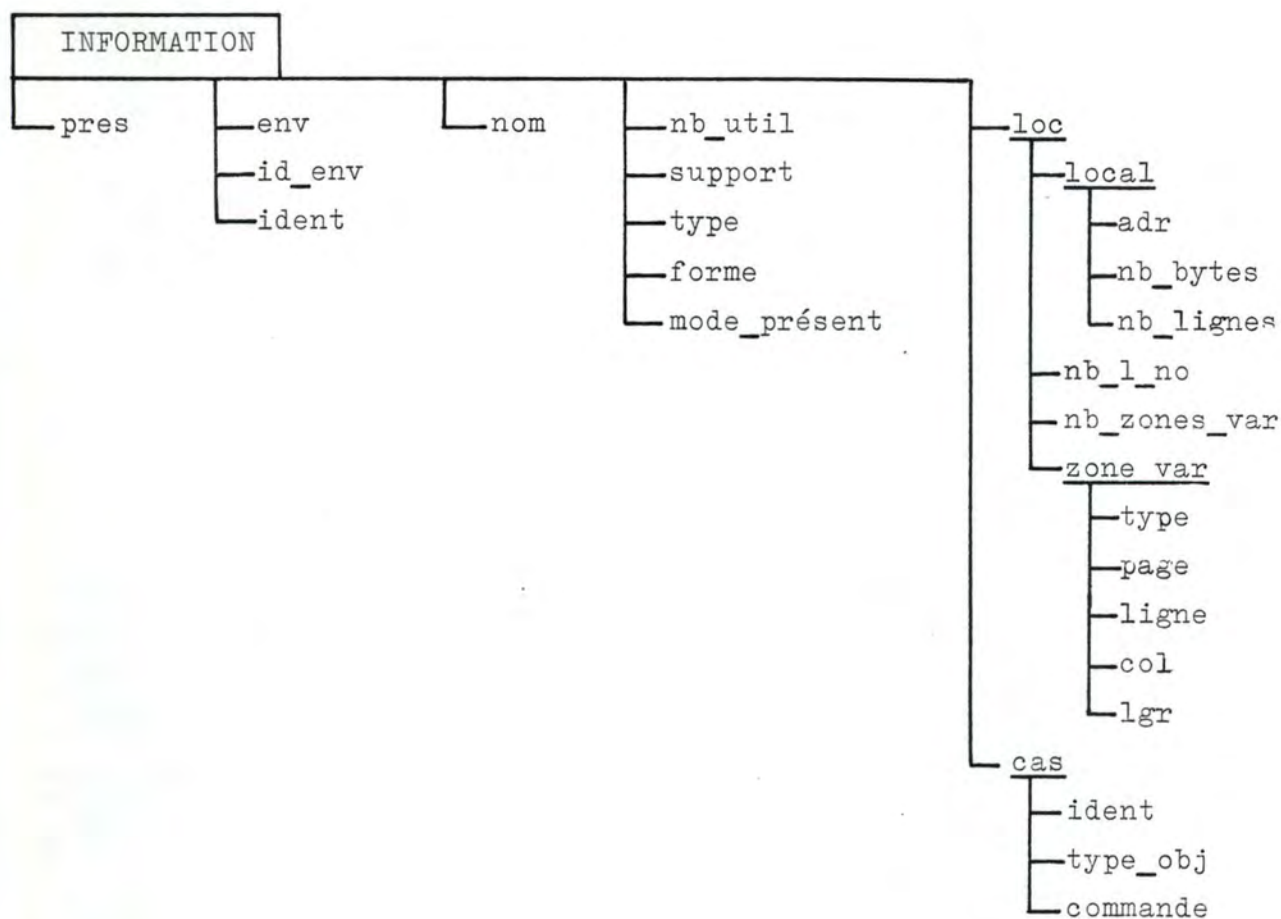
identifiante

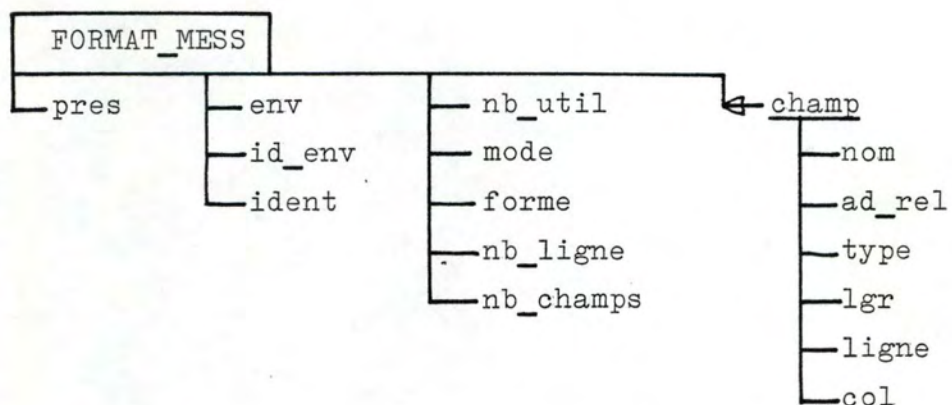


clés

primaire env, id_env, ident

secondaire env, id_env, nom

clésprimaire env, id_env, identsecondaireidentifiante env, id_env, nom

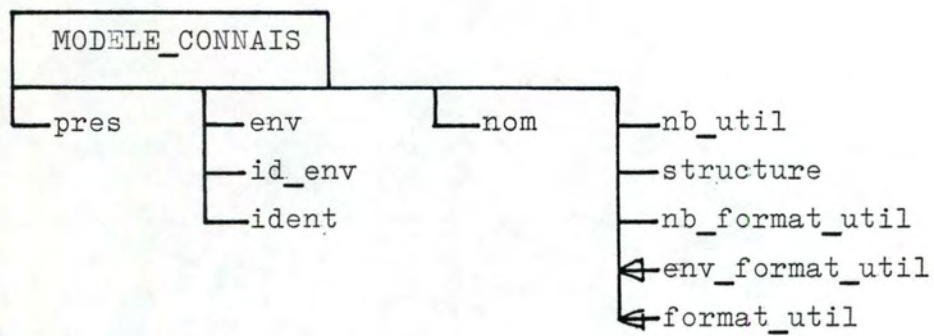


clés

primaire env, id_env, ident

secondaire

identifiante env, id_env, nom

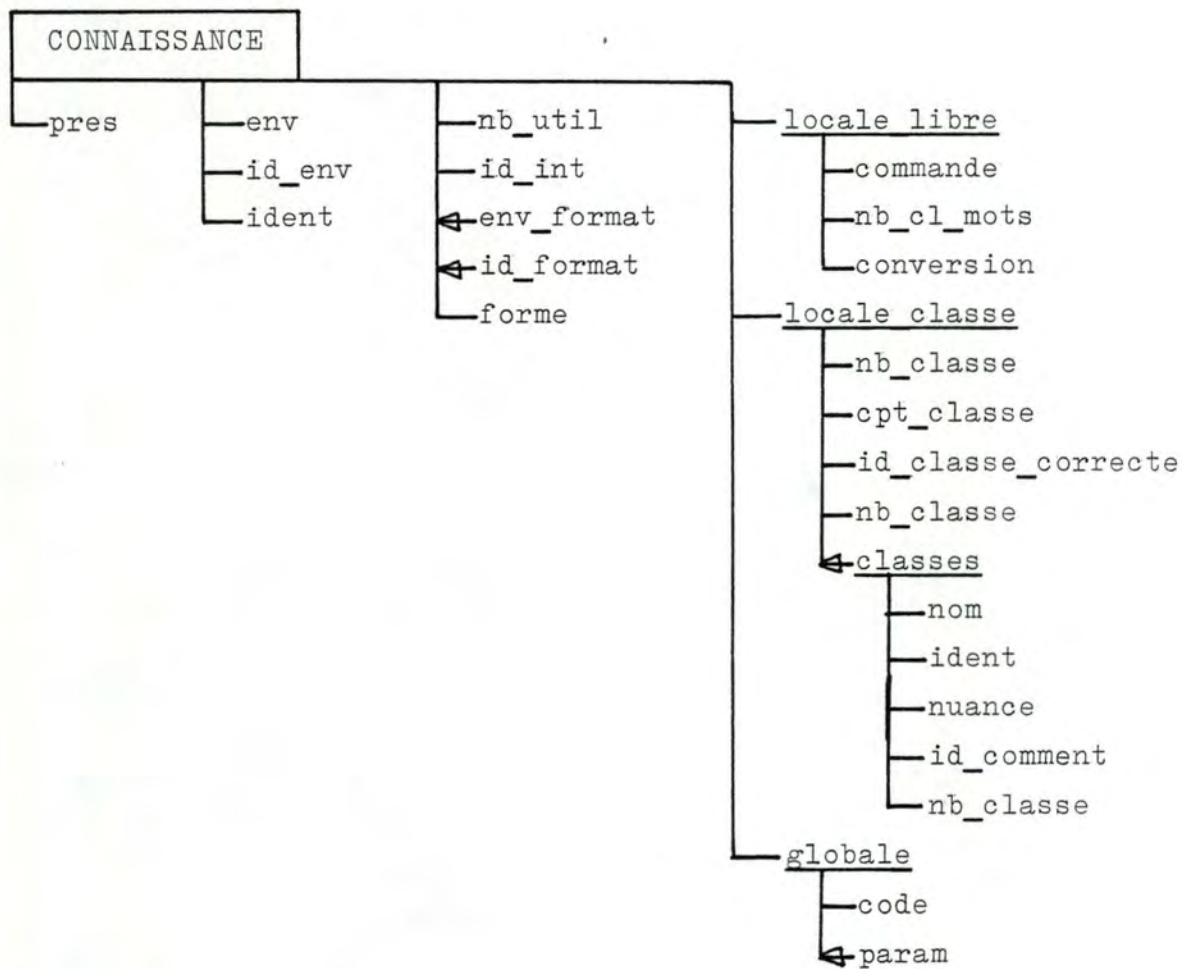


clés

primaire env, id_env, ident

secondaire

identifiante env, id_env, nom

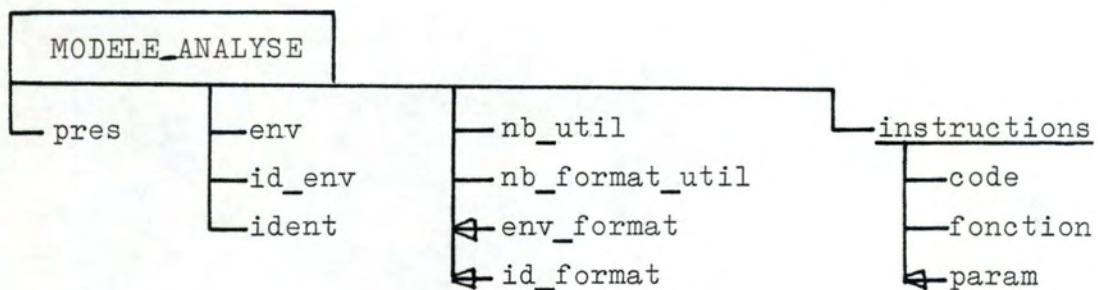


clés

primaire env, id_env, ident

secondaire

identifiante env, id_env, nom

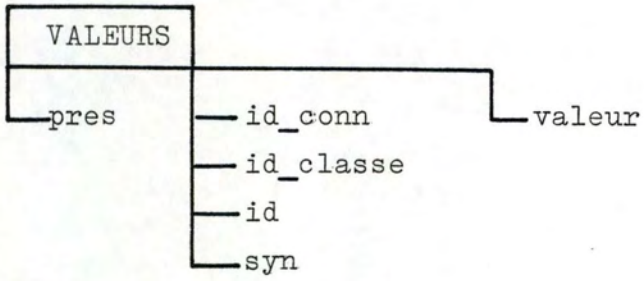


clés

primaire env, id_env, ident

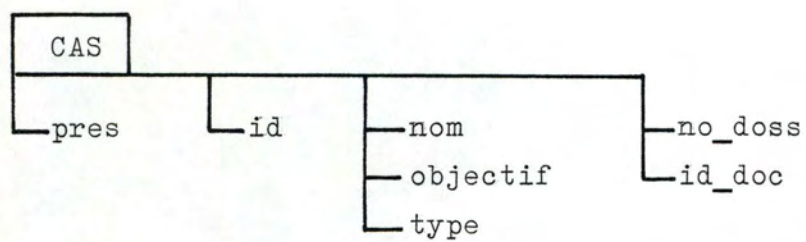
secondaire

identifiante env, id_env, nom



clé

primaire id_conn, id_classe, id, syn



clés

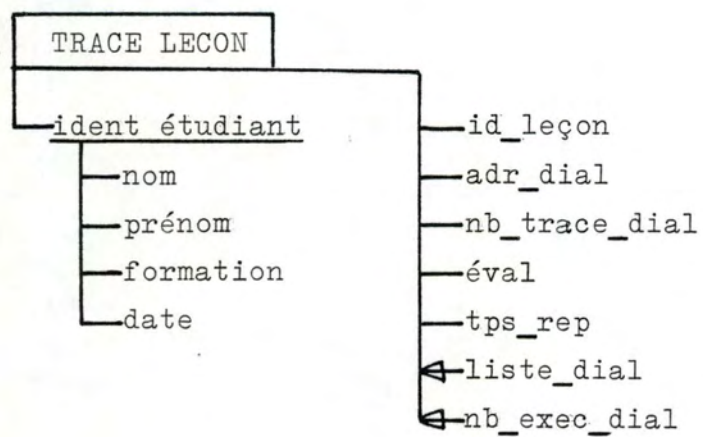
primaire id

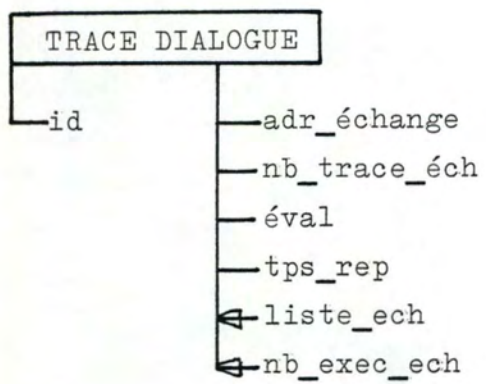
secondaires

identifiante nom

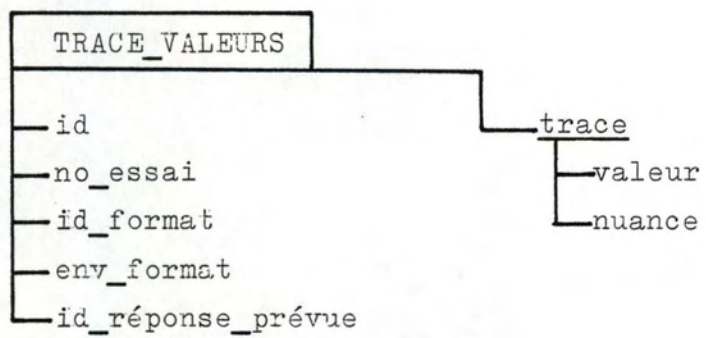
multiples objectif

type





TRACE_ECHANGE	
—id	—adr_val
—no_essai	—nb_trace_val
	—éval
	—tps_rep
	—nb_correctes
	—nb_erreurs
	—nb_tolérables
	—nb_non_reconn
	—nb_manquantes
	—nb_redondantes
	—nb_incomplètes
	—type_erreur
	◀requete
	◀consult



CONCLUSION

Le but de ce mémoire a été de concevoir un outil assez souple et modulaire pour s'adapter à la diversité des structures de connaissance des modèles de décision associés au domaine de l'enseignement de la discipline homéopathique.

La conception des leçons, en termes d'objets pédagogiques permet d'étendre le noyau du système proposé à d'autres objets, ou à paramétrer les objets existants.

Le suivi des leçons, abordé seulement dans le présent projet, permettrait de réajuster le processus d'enseignement en relevant les types d'erreurs caractéristiques. Cette démarche permettrait également d'enrichir les environnements de travail servant de base à l'orientation des dialogues.

Une optimisation des accès fichiers, lorsqu'une leçon est définitivement acceptée, diminuerait le temps de réponse à l'exécution.

A N N E X E

STRUCTURES AUXILIAIRES

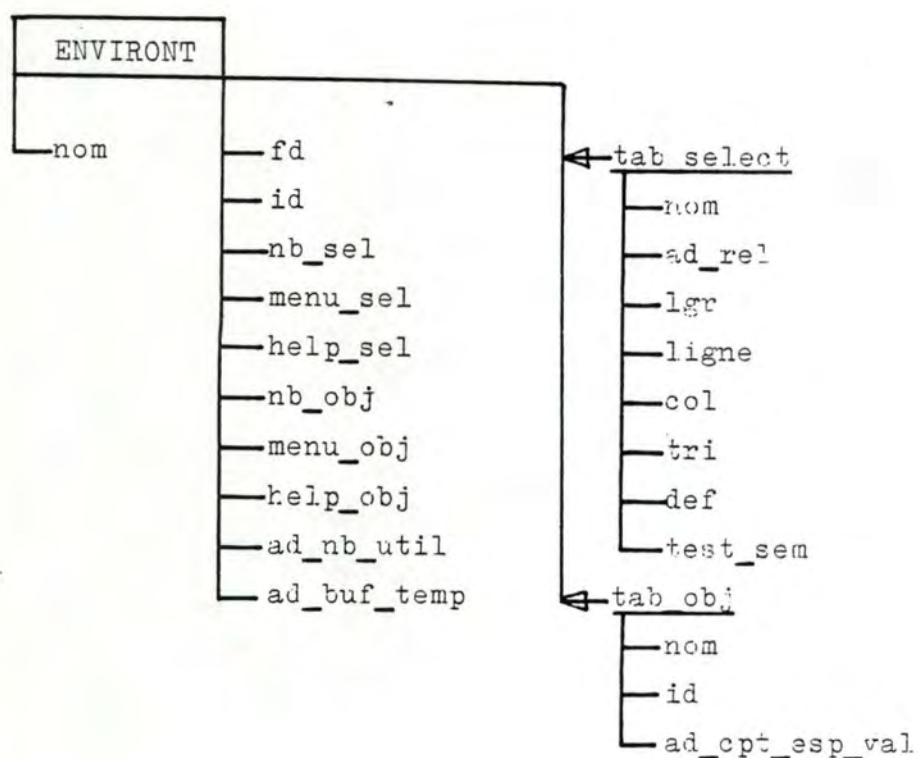
- Structure supportant le graphe des environnements de validité.

- A chaque élément de cette structure correspond un objet.
- Chaque objet est décrit par :
 - . fd : l'identifiant du fichier correspondant
 - . id : l'identifiant du type de l'objet
 - . nb_sel : nombre de critères de sélection
 - . menu_sel : menu des attributs sur lesquels la sélection de l'objet peut être effectuée.
 - . help_sel : écran d'aide associé au menu de sélection
 - . nb_obj : nombre de sous-objets valides dans l'objet (considéré comme environnement) courant.
 - . menu_obj : menu des noms des sous-objets valides dans l'objet courant.
 - . help_obj : écran d'aide associé au menu des sous-objets.
 - . ad_nb_util : adresse relative, dans un record de l'objet, de l'attribut nb_util.
 - . ad_buf_temps : adresse du tampon temporaire associé au fichier de l'objet.
 - . Pour chaque critère de sélection :
 - . nom : nom de l'attribut servant de critère de sélection.
 - . ad_rel : adresse relative, dans un record de l'objet, de l'attribut déterminé par nom.
 - . lgr : longueur maximale de la valeur de l'attribut dans un record
 - . ligne } : ligne et colonne déterminant la position à
 - . col } : l'écran, à laquelle devra être affiché l'attribut lors de la sélection de l'objet.
 - . tri : indicateur déterminant si l'attribut fait partie de la clé de tri.
 - . def : valeur par défaut de l'attribut.
 - . test_sem : fonction testant la sémantique de la valeur de l'attribut lors de sa création ou modification.

. Pour chaque sous-objet :

- . nom : nom du sous-objet.
- . id : identifiant du type de sous-objet
- . ad_cpt_esp_val :
adresse relative, dans un record de l'objet,
du compteur des sous-objets de type id
valides dans l'environnement objet.

- schéma d'un record du fichier correspondant à cette structure :



- Structures décrivant les environnements temporaires lors

de l'exécution.

- C'est sur base des éléments de ces environnements de travail que sont évaluées les règles de chaînage entre échanges ou dialogues.
- Chaque élément de ces environnements est décrit par :
 - . nom : nom de l'objet sur lequel peut porter une condition.
 - . adresse : adresse en mémoire de l'objet correspondant
 - . type : le type de valeur associée à cet objet.
 - . nature : la nature (élémentaire ou tableau) de l'objet.
 - . dim : adresse du compteur déterminant la dimension de l'objet si celui-ci est de nature tableau.
- Description de l'environnement env_leçon.

Nom	adresse	type	nature	dim
évaluation	& éval_lec	n	e	-
temps_rep	& tps_lec	r	e	-
liste_dial	liste_d	n	t	& cpt_d
nb_exec_dial	nb_x_d	n	t	& cpt_d

- Description de l'environnement env_dialogue.

Nom	adresse	type	nature	dim
évaluation	& éval_d	n	e	-
temps_réponse	& tps_d	r	e	-
liste_échange	liste_c	n	t	& cpt_e
nb_exec_ech	nb_x_e	n	t	& cpt_e

- Description de l'environnement env_échange

Nom	adresse	type	nature	dim
nb_correctes	& nb_c	n	e	-
nb_erronées	& nb_e	n	e	-
nb_non_reconnus	& nb_n	n	e	-
nb_manquantes	& nb_m	n	e	-
nb_tolérables	& nb_t	n	e	-
nb_redondantes	& nb_r	n	e	-
nb_incomplètes	& nb_i	n	e	-
type_erreur	& type_er	n	e	-
nb_essai	& nb_es	n	e	-
évaluation	& eval_e	n	e	-
tps_rep	& tps_e	r	e	-
requête	req	n	t	& cpt_req
consult	cons	n	t	& cpt_cons

B I B L I O G R A P H I E

1. T.F.A. ALLEN
The Encyclopedia of Pure Materia Medica (12 vol.)
B.Jain Publishers. Nex Delhi - 110016 . 1977
2. P. BERTIER, J.M. BOUROCHE
Analyse des données multidimensionnelles
P.U.F., Paris, 1975
3. H. BESTOUGEFF, J.-P. FARGETTE
Enseignement et ordinateur
mémoire présenté aux F.N.D.P. - Institut d'Informatique, 1980
4. F.M. BLONDEL et al
Enseignement assisté par ordinateur en médecine; expérience
en hématologie
Computers in Education, O. Lecarme and R. Lewis, ids, IFIP
1975, North-Holland
5. F. BODART, Y. PIGNEUR
Conception assistée des applications informatiques. 1.étude
d'opportunité et analyse conceptuelle
Ed. Masson, Paris, 1983
6. P. BORGEVIN et al
Un système d'aide à l'enseignement du diagnostic médical
Actes de la IXe Ecole Internationale d'Eté d'Informatique de
l'AFCET, Monastir, 10-22 juillet 1979
7. C. CLARENNE, C. SONNET
Réalisation et exploitation d'une base de dossiers cliniques
mémoire présenté aux F.N.D.P., Namur - Institut d'Informatique
1984
8. H. de MONTMOLLIN
L'enseignement programmé
P.U.F. - Coll. Que sais-je? - Paris, 1975 4e édition
9. P. DEROY
Aide au diagnostic médical. Synthèse des méthodes et applications
Mémoire présenté aux F.N.D.P., Namur - Institut d'Informatique
1979
10. J. DONNAY, H. ROMAINVILLE
Grille d'analyse des didacticiels
réseau O.S.C., Centre d'Information sur l'Apprentissage et
l'Enseignement assisté par ordinateur, 1984
11. J. FICHEFET
Aide à la décision et diagnostic médical assisté par ordinateur
Actes de la IXe Ecole Internationale d'Eté d'Informatique de
l'AFCET, Monastir, 10-22 juillet 1979, pp339-352
12. J. FICHEFET
Une banque de schémas diagnostiques en homéopathie?
Esquisse provisoire de propositions.
Rapport interne U.I.H.N., mai 1983

13. S.C.F. HAHNEMANN
Organon der Heilkunst
6 Auflage, Leipzig, Verlag von Dr. Willmar Schwabe, 1921
14. C. HERING
The Guiding Symptoms of our Materia Medica (10 vol.)
B. Jain Publishers - 1-55, Arjun Nagar. New Delhi - 16. 1971
15. J. HUI BON HOA
Précis de technique répertoriale homéopathique de Kent
Ed. Coquemard, Angoulême, 1963
16. A. JACQUES
Introduction à l'homéopathie hahnemaniennne
rapport U.I.H.N., Namur, 1983
17. A. JACQUES
Première prescription. Deuxième prescription. Pronostic -
Lois de Hering
rapport U.I.H.N., Namur, 1980
18. A. JACQUES, J. FICHEFET, P. GARDIN, J. PARIS
Homéopathie, cybernétique et aide à la décision multicritère
Namur, 1983
19. T. KENNETH
The building of a computer based teaching system
Computers in Education, Vol. 8, n°1 pp15-19, 1984
20. J.T. KENT
Repertory of the Homeopathic Materia Medica
Ed. Ehrart et Karl, Chicago
21. H. MATELART
Vers une évaluation de la capacité de déduction logique dans
un système multimedia d'enseignement - Les système IMAGO
CEDIC, Fernand Nathan, Paris, 1982
22. J.M. ROGER, P. LENOIR
Un système d'aide au diagnostic médical, l'ADM
Actes de la IXe Ecole Internationale d'Eté Informatique de
l'AFCET, Monastir, 10-22 juillet 1979, pp 362-374
23. P. SCHMIDT
La consultation homéopathique. L'art d'interroger
Imprimerie Rasp, Genève, 1975
24. R. SIMONIN et al
Enseignement de la pathologie thyroïdienne par ordinateur
Computers in Education
25. R. SWARTOUT
A digitalis therapy advisor with explanations
Submitted to the departement of electrical engineering
and Computer Science, Januari 16, 1977, M.I.T.
26. J. WHITING
Cognitive and student assessments of a CAC package designed
for mastery learning
Computers in Education, Vol 8 n°1 1984, pp59-67